

147 lines (79 loc) · 4.59 KB

T1071.001 - Web Protocols

Description from ATT&CK

Adversaries may communicate using application layer protocols associated with web traffic to avoid detection/network filtering by blending in with existing traffic. Commands to the remote system, and often the results of those commands, will be embedded within the protocol traffic between the client and server.

Protocols such as HTTP and HTTPS that carry web traffic may be very common in environments. HTTP/S packets have many fields and headers in which data can be concealed. An adversary may abuse these protocols to communicate with systems under their control within a victim network while also mimicking normal, expected traffic.

Atomic Tests

- [Atomic Test #1 - Malicious User Agents - Powershell](#)
- [Atomic Test #2 - Malicious User Agents - CMD](#)

- [Atomic Test #3 - Malicious User Agents - Nix](#)

Atomic Test #1 - Malicious User Agents - Powershell

This test simulates an infected host beaconing to command and control. Upon execution, no output will be displayed. Use an application such as Wireshark to record the session and observe user agent strings and responses.

Inspired by APTSimulator - <https://github.com/NextronSystems/APTSimulator/blob/master/test-sets/command-and-control/malicious-user-agents.bat>

Supported Platforms: Windows

auto_generated_guid: 81c13829-f6c9-45b8-85a6-053366d55297

Inputs:

Name	Description	Type	Default Value
domain	Default domain to simulate against	String	www.google.com

Attack Commands: Run with `powershell` !

```
Invoke-WebRequest #{domain} -UserAgent "HttpBrowser/1.0" | out-null
Invoke-WebRequest #{domain} -UserAgent "Wget/1.9+cvs-stable (Red Hat modified)" | out-null
Invoke-WebRequest #{domain} -UserAgent "Opera/8.81 (Windows NT 6.0; U; en)" | out-null
Invoke-WebRequest #{domain} -UserAgent "*<|>*" | out-null
```

PreviewCodeBlame

RawCopyDownloadMenu

Atomic Test #2 - Malicious User Agents - CMD

This test simulates an infected host beaconing to command and control. Upon execution, no out put will be displayed. Use an application such as Wireshark to record the session and observe user agent strings and responses.

Inspired by APTSimulator - <https://github.com/NextronSystems/APTSimulator/blob/master/test-sets/command-and-control/malicious-user-agents.bat>

Supported Platforms: Windows

auto_generated_guid: dc3488b0-08c7-4fea-b585-905c83b48180

Inputs:

Name	Description	Type	Default Value
domain	Default domain to simulate against	String	www.google.com
curl_path	path to curl.exe	Path	C:\Windows\System32\Curl.exe

Attack Commands: Run with **command_prompt** !

```
{curl_path} -s -A "HttpBrowser/1.0" -m3 #{domain} >nul 2>&1
{curl_path} -s -A "Wget/1.9+cvs-stable (Red Hat modified)" -m3 #{domain} >nul 2>&1
{curl_path} -s -A "Opera/8.81 (Windows NT 6.0; U; en)" -m3 #{domain} >nul 2>&1
{curl_path} -s -A "*<|>*" -m3 #{domain} >nul 2>&1
```

Dependencies: Run with **powershell** !

Description: Curl must be installed on system

Check Prereq Commands:

```
if (Test-Path {curl_path}) {exit 0} else {exit 1}
```

Get Prereq Commands:

```
Invoke-WebRequest "https://curl.haxx.se/windows/dl-7.71.1/curl-7.71.1-win32-mingw..
Expand-Archive -Path $env:temp\curl.zip -DestinationPath $env:temp\curl
Copy-Item $env:temp\curl\curl-7.71.1-win32-mingw\bin\curl.exe {curl_path}
Remove-Item $env:temp\curl
Remove-Item $env:temp\curl.zip
```

Atomic Test #3 - Malicious User Agents - Nix

This test simulates an infected host beaconing to command and control. Inspired by APTSimulator - <https://github.com/NextronSystems/APTSimulator/blob/master/test-sets/command-and-control/malicious-user-agents.bat>

Supported Platforms: Linux, macOS

auto_generated_guid: 2d7c471a-e887-4b78-b0dc-b0df1f2e0658

Inputs:

Name	Description	Type	Default Value
domain	Default domain to simulate against	String	www.google.com

Attack Commands: Run with `sh` !

```
curl -s -A "HttpBrowser/1.0" -m3 #{domain}
curl -s -A "Wget/1.9+cvs-stable (Red Hat modified)" -m3 #{domain}
curl -s -A "Opera/8.81 (Windows NT 6.0; U; en)" -m3 #{domain}
curl -s -A "*<|>*" -m3 #{domain}
```

