



NetRipper – Smart traffic sniffing for penetration testers

Ionut Popescu – Senior Application Security Engineer @ 1&1 Romania



About me

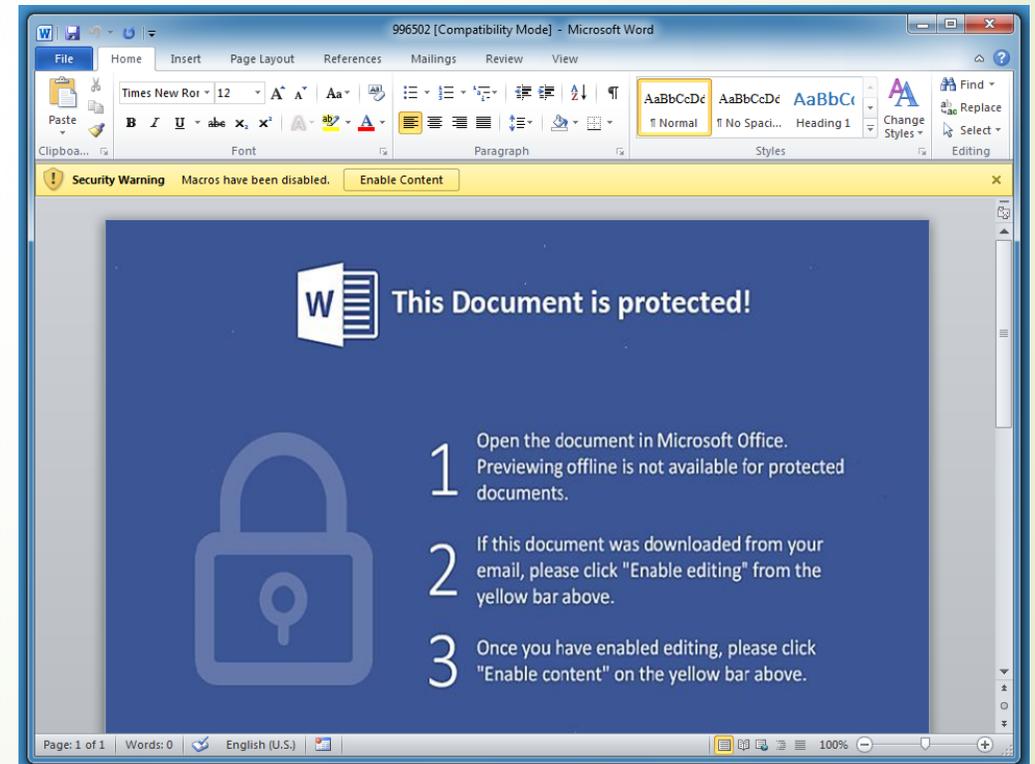
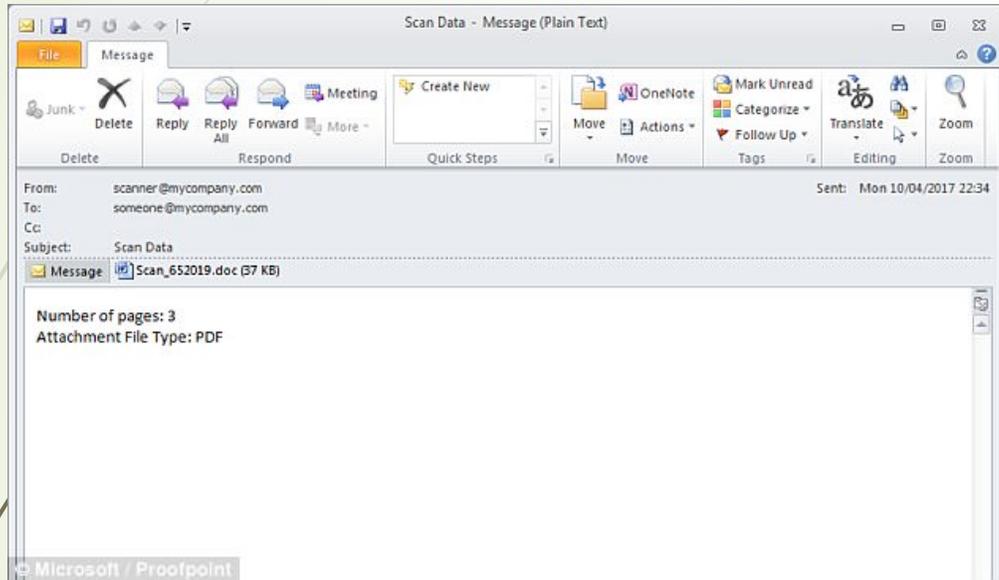
- Blogger @ <https://nytrosecurity.com/>
- GitHub @ <https://github.com/NytroRST>
- Twitter @ <https://twitter.com/NytroRST>
- Admin @ <https://rstforums.com/forum/>



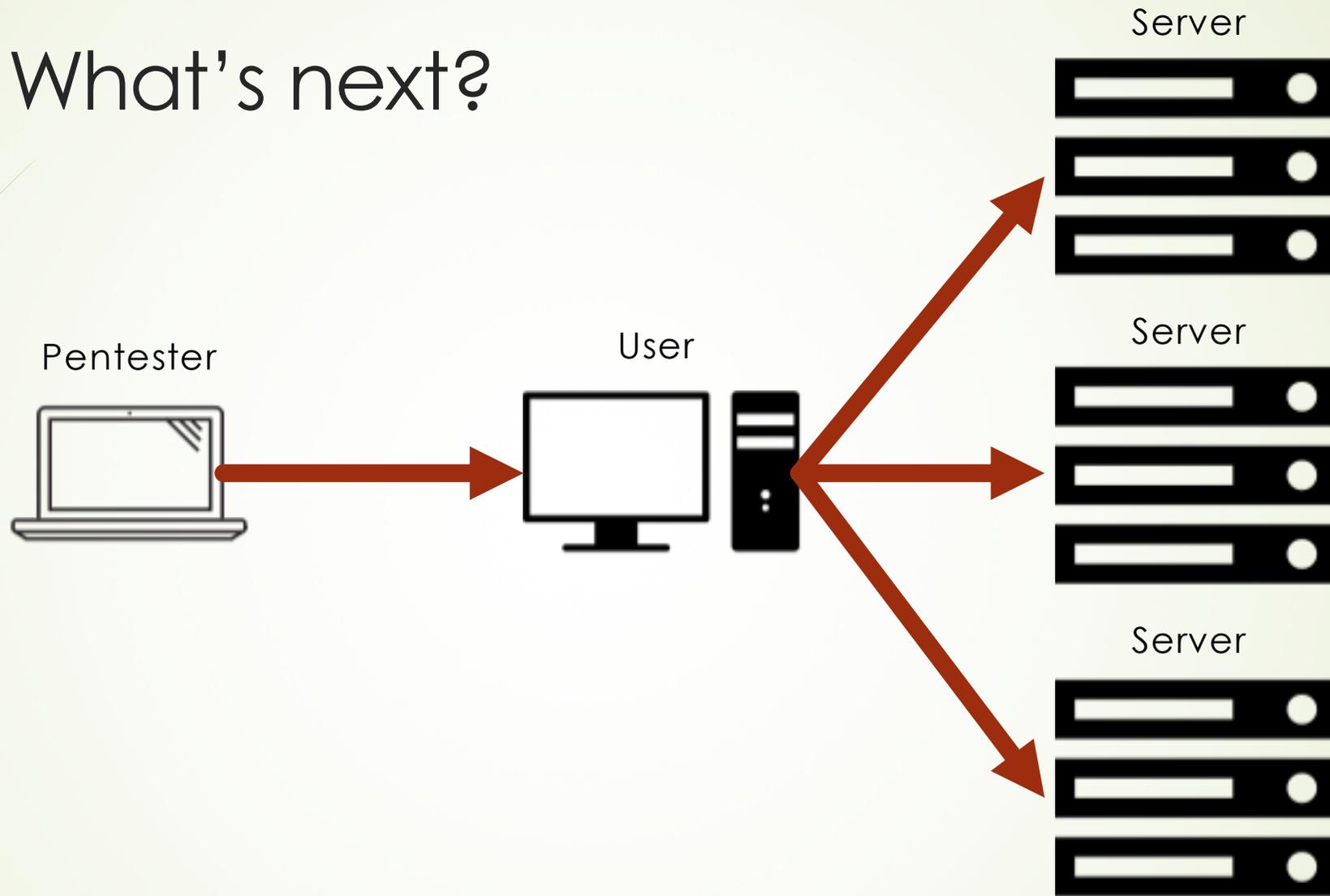
NetRipper

- Personal project
- Released at Defcon 23 (2015)
- Presented at BlackHat Asia Arsenal (2018)
- For penetration testers
- For anyone

Getting access to a workstation



What's next?



How to connect to servers?

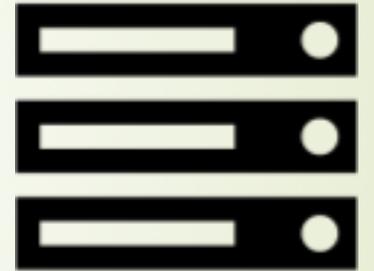


Connection example



SSH: 22

596f752077617374656420796f75722074696d65



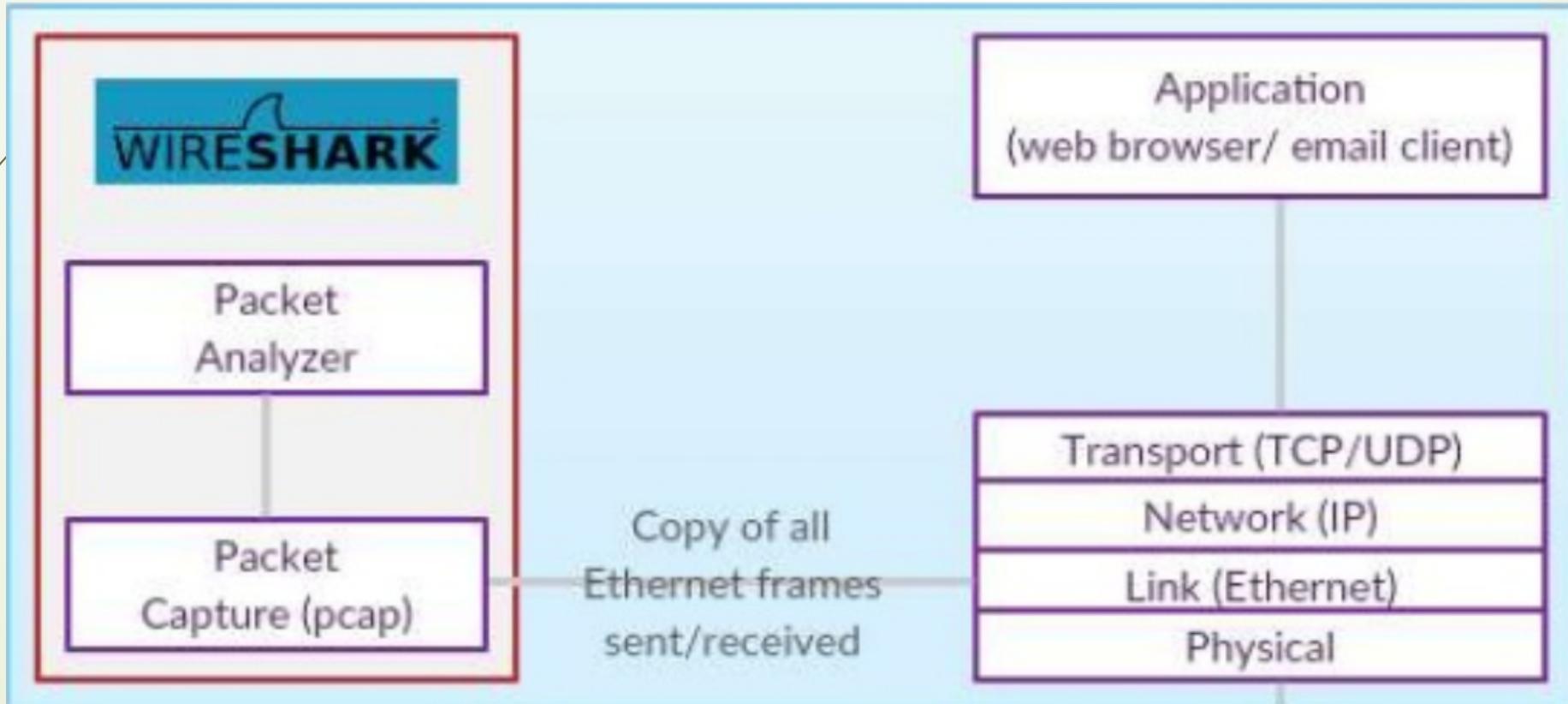
Traffic sniffing

```
Wireshark · Follow TCP Stream (tcp.stream eq 3) · wireshark_8E7802C7-1657-431E-A03D-345E41FF3F2D_20180425183603_a01572.pcapng

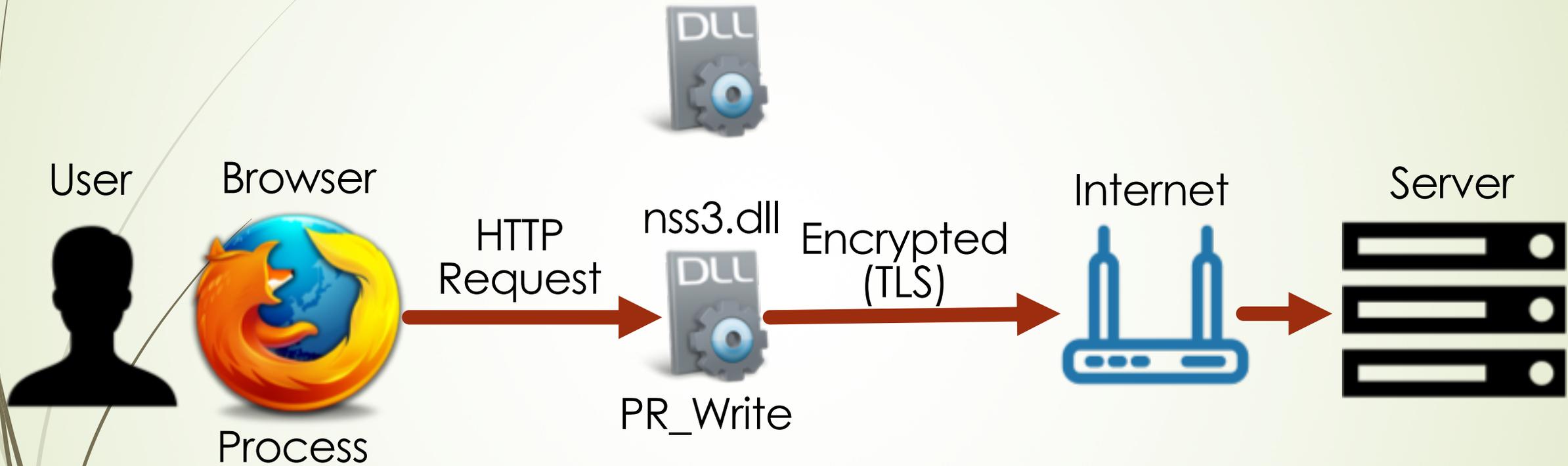
SSH-2.0-PuTTY_Release_0.70
SSH-2.0-OpenSSH_7.4p1 Debian-10+deb9u3
...L...<.;.l.lChF...curve25519-sha256@libssh.org,ecdh-sha2-nistp256,ecdh-sha2-nistp384,ecdh-sha2-nistp521,diffie-hellman-group-exchange-
sha256,diffie-hellman-group-exchange-sha1,diffie-hellman-group14-sha1,rsa2048-sha256,rsa1024-sha1,diffie-hellman-group1-sha1...Wssh-ed25519,ecdsa-sha2-
nistp256,ecdsa-sha2-nistp384,ecdsa-sha2-nistp521,ssh-rsa,ssh-dss...aes256-ctr,aes256-cbc,rijndael-cbc@lysator.liu.se,aes192-ctr,aes192-cbc,aes128-ctr,aes128-
cbc,chacha20-poly1305@openssh.com,blowfish-ctr,blowfish-cbc,3des-ctr,3des-cbc,arcfour256,arcfour128...aes256-ctr,aes256-cbc,rijndael-
cbc@lysator.liu.se,aes192-ctr,aes192-cbc,aes128-ctr,aes128-cbc,chacha20-poly1305@openssh.com,blowfish-ctr,blowfish-cbc,3des-ctr,3des-
cbc,arcfour256,arcfour128...hmac-sha2-256,hmac-sha1,hmac-sha1-96,hmac-md5,hmac-sha2-256-etm@openssh.com,hmac-sha1-etm@openssh.com,hmac-sha1-96-
etm@openssh.com,hmac-md5-etm@openssh.com...hmac-sha2-256,hmac-sha1,hmac-sha1-96,hmac-md5,hmac-sha2-256-etm@openssh.com,hmac-sha1-etm@openssh.com,hmac-sha1-96-
etm@openssh.com,hmac-md5-etm@openssh.com...none,zlib...none,zlib.....X....4.....!.....c.S....curve25519-sha256,curve25519-
sha256@libssh.org,ecdh-sha2-nistp256,ecdh-sha2-nistp384,ecdh-sha2-nistp521,diffie-hellman-group-exchange-sha256,diffie-hellman-group16-sha512,diffie-hellman-
group18-sha512,diffie-hellman-group14-sha256,diffie-hellman-group14-sha1...Assh-rsa,rsa-sha2-512,rsa-sha2-256,ecdsa-sha2-nistp256,ssh-ed25519...lchacha20-
poly1305@openssh.com,aes128-ctr,aes192-ctr,aes256-ctr,aes128-gcm@openssh.com,aes256-gcm@openssh.com...lchacha20-poly1305@openssh.com,aes128-ctr,aes192-
ctr,aes256-ctr,aes128-gcm@openssh.com,aes256-gcm@openssh.com...umac-64-etm@openssh.com,umac-128-etm@openssh.com,hmac-sha2-256-etm@openssh.com,hmac-sha2-512-
etm@openssh.com,hmac-sha1-etm@openssh.com,umac-64@openssh.com,umac-128@openssh.com,hmac-sha2-256,hmac-sha2-512,hmac-sha1...umac-64-etm@openssh.com,umac-128-
etm@openssh.com,hmac-sha2-256-etm@openssh.com,hmac-sha2-512-etm@openssh.com,hmac-sha1-etm@openssh.com,umac-64@openssh.com,umac-128@openssh.com,hmac-
sha2-256,hmac-sha2-512,hmac-sha1...none,zlib@openssh.com...none,zlib@openssh.com...Q...4s...?.iKs...0.=...`W.P..
3.....3.....ssh-ed25519...83..(....5k.....o.B.XH.....Z.....h.c.e.x.&.._j...P.....S.....ssh-
ed25519...@..f!>...t.e.r...P.....IT...J...^aRw..RL4.FF)...$\\.....A.._
.....
.....
...OR>.!...|.....S.....y.....00...:r...9....(+3k=~@.
...\.o.o?>...<.q..7..Mb...D.d..ll:K...PvR..._2...rP...{s...S.m.*0E.#. /./1jt1}...|%.h4....?R.-v.../a..s.%. _Y...
.Qx.s.....[...#.k...I.N1...+...l.f.,)...x.....IJ...^1.9.VJ.....>
n...tPI..%
Ha../ADaw.../NAV=.....;33v5M...V...~.'KY.9.z.....c~.....RZ...E.....+..._4e..W."y.\.t/5l.d...g.....
...l..vP.u...j.....6n.e..U.>[.)
..{t7...*5....4..C*.i:./.,XWk8f..fXiB...9@.&`.....pR('.<n~...yn.Z^..k.....S.h.....z0...3....qPX...A..=v
\55...J...K,.e.....By.....g..y.oh...Jw..?.8p.NQ).....H.....#...-O.d&l.-b..V..Ox.....a..a.&..I...!K.p. ....i.....o@.....'.....n].
4.D...'.T%... ..p.t./.....{.6.....}.idp..3+ ja
.4....Qd.T~.....c.V.Qf...+...N<t.....a\.....#..0.....P9z.....Z.....m..8.....Wa.|$.I..b..
.Dj;...S..Nl..vt.D...pf.....
..z...9sz[.2..xzbx?.|.2.q..Z.f...-#Z:.-?L...T4:?.?..wh.C...WN(=%....%.1.....t.
...p...
gq.jf.n...F...~@I.8...a.o.e...l.q.;.x...?.. ..b..9.Q..i.....mKo..k.][...v.^..Z.$U..1.`.OL'...bL.....1`..o;/I.kT.....
3.*.N.v2.....^
9|y...87.B...>.p;....87.B...>.p;....Ab.*.II4.....F...z.j.R.M.w{).%.*Uc\.....S...FQH:.#.....?e...6(. FZ"U.....D2i/.V...^..L.u.F.
.R.....+1.../..@.4'..|.....z]..w.P...} ..U.O1Q..|....ky.[T..O...E9*.p6w. ....F.<J.....6").u.].7[.;?
D.....0.<...<.....}.....v.T...d...j.....:0X.%y..}"}..l&.V3..._...
.Sn.M@.v.Y...#jT~
b.w.\Ba.IW..J.g.;X[-.t.m+.....3.n;Z..3?..?....
"E.I...tR..r.Vw.....Zl:..G..h.{.PY..E..m.w..JWsk8~.#.}|...&... ..q>u...~FY...;X.6.....FRC.+...?..+.....oe..b./R.....1.....\sw.
g...F.e.g...}. ....Po.: 'y4...R...I..&.)Y#...w.A..gh.DG.l+K.5JG,....~$HA.o(.30...j${.....8.l\...P ..q..h)...&..&=..c...uL..Q.us.U
%...N...>...3.6...k+.g.Z.u.<3.u.A...Y...7...U..10.{.....@ .A[.....Zh.lL.u.HI.....)0.^..W.....W.f..R
```

How a sniffer works

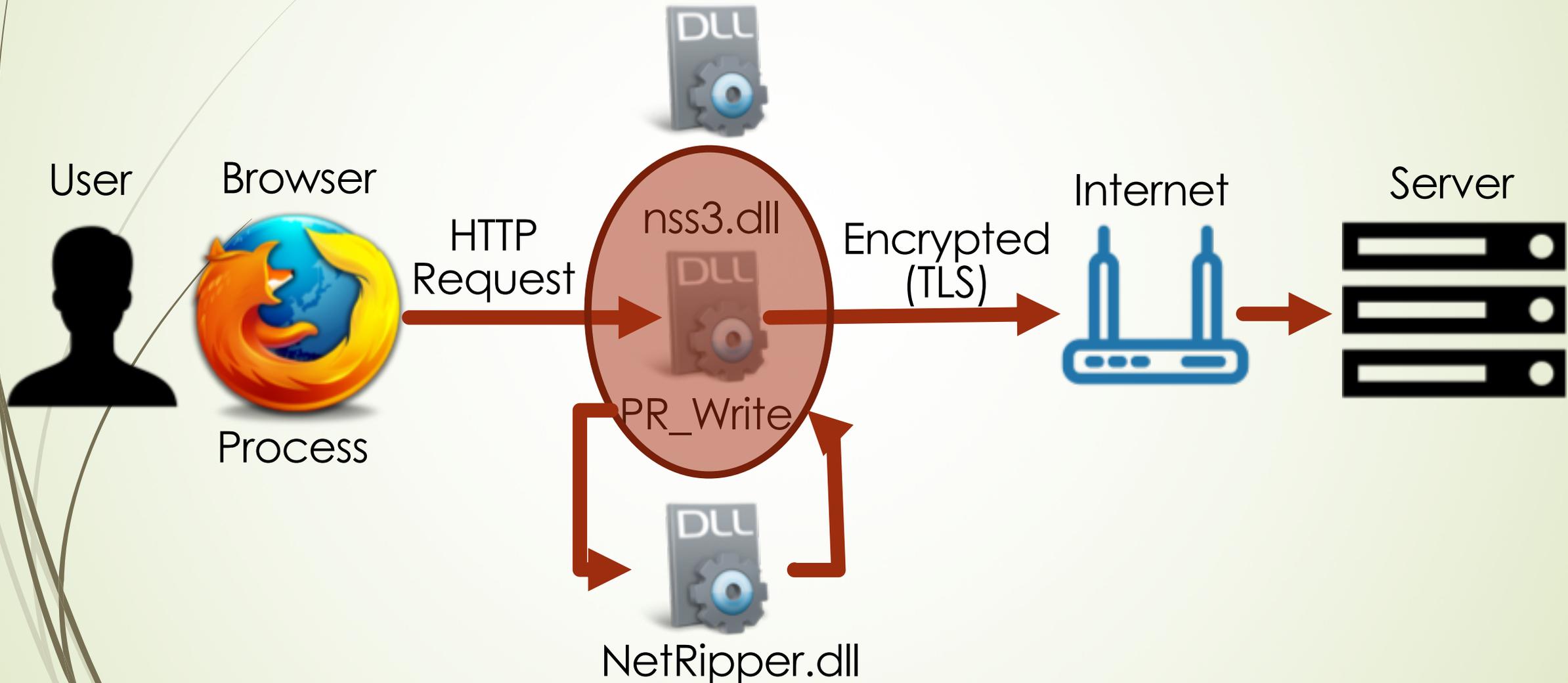
- It requires administrative privileges
- Useless for encrypted data (e.g. HTTPS, SSH)



How a browser works



How NetRipper works



API Hooking

GET / HTTP/1.1
Host: mail.com



Encrypted data

Encrypted data



GET / HTTP/1.1
Host: mail.com



NetRipper components

- NetRipper.dll – Main component (API hooking)
- NetRipper.exe – DLL configurator and injector
- netripper.rb – Metasploit module of DLL configurator and injector



NetRipper plugins

- PlainText – Save only plaintext data
- DataLimit – Limit „packet” size
- StringFinder – Find strings



What's new?

- Cross-compilation on Linux
 - Support for PCAP files
- 



Cross-compilation on Linux

- Requires mingw-w64
 - Compiled DLLs are big
 - Has limitations
 - Will be improved
- 

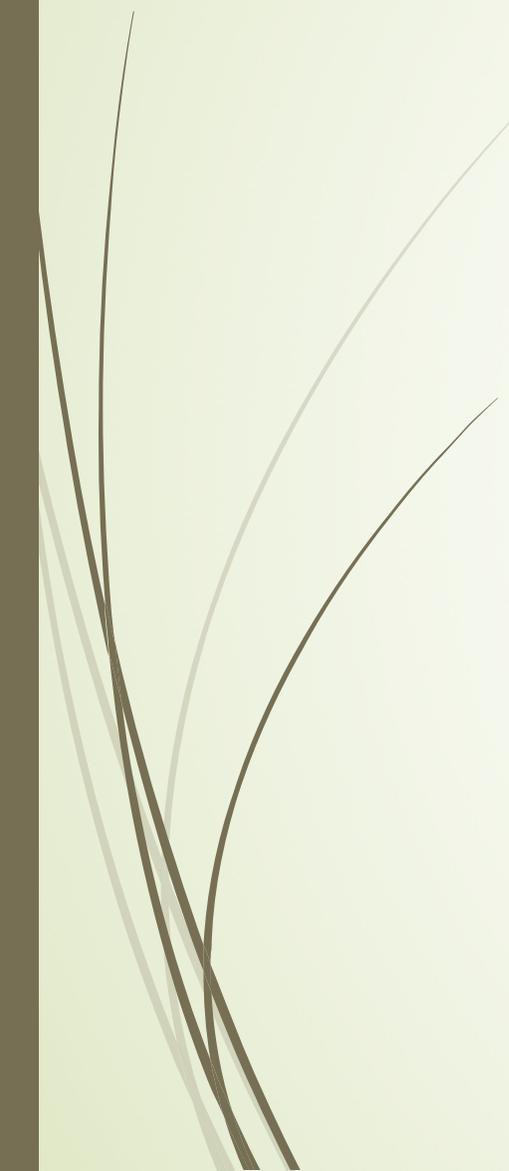


PCAP files

- Can easily follow requests and responses
- Can be used with Wireshark (supports multiple protocols)
- Can be used with other tools supporting PCAP files
- Can get IP addresses and TCP ports (limited)
- Will be improved



Where is the problem?

- Some applications are statically linked (no exported functions, reverse-engineering required)
 - Support has to be added for each of them
 - Examples: Putty, Google Chrome
- 

Google Chrome

The screenshot displays the Immunity Debugger interface with the following components:

- Disassembly Window:** Shows assembly code for the function `chrome.dll:$1CC2F5E #1CC235E`. The instruction `call chrome.6A90D998` is highlighted with a red box.
- Context Menu:** A menu is open over the assembly view, listing various actions such as Binary, Copy, Breakpoint, Follow in Dump, Follow in Disassembler, Follow in Memory Map, Decompile, Graph, Help on mnemonic, Show mnemonic brief, Highlighting mode, Label, Trace record, Comment, Toggle Bookmark, Analysis, Assemble, Patches, Yara..., Set New Origin Here, Create New Thread Here, Go to, Search for, and Find references to.
- Register Window (Right):** Shows the state of registers: EAX (0C71FCE4), EBX (00000000), ECX (00000004), EDX (00000000), EBP (0C71FD0C), ESP (0C71FCA0), ESI (00000BEC), EDI (0C71FCE8), and EIP (77D6F901). It also shows EFLAGS (00000246) and the last error status (ERROR_SUCCESS).
- Memory Dump Window (Bottom):** Shows a hex dump of memory with ASCII characters below it, including the string `./third_party/boringssl/src/ssl/`.



Use cases

- Penetration testers
 - Bug bounty hunters
 - Attackers
 - Any other users
- 



DEMO



Improvements

- Support for multiple applications
- Performance and stability
- Bypass process mitigations
- Inject in new processes
- More plugins (e.g. regular expressions)
- Support for Linux/Mac?



Conclusion

- Open-source tool for Windows
 - Captures traffic before encryption and after decryption
 - Supports multiple applications
 - Easy to use
 - It can be improved
- 



Questions?

<https://github.com/NytroRST/NetRipper>

ionut.popescu@outlook.com