

BEHIND THE CURTAIN

INFORMATIKUSOK VS. PENTESTEREK



Veres-Szentkirályi András 2021-02-27



Veres-Szentkirályi András

- ▶ OSCP, GWAPT, SISE
- ▶ Silent Signal alapító
- ▶ pentester, toolmaker

- ▶ bizonyos módszertan mellett minden hibát feltár
- ▶ nem csak első vérig (vö. bug bounty)

Eldobható kód



- ▶ Python, Ruby, Perl
- ▶ REPL: IPython
- ▶ CTF külön kategória
- ▶ gyors fejlesztés vs. gyors futás, olvashatóság, karbantarthatóság
- ▶ instrumentálhatóság vs. specifikáció követése

Egyedi protokollok kezelése



```
decode_packets(<<?OLD_PACKET_FORMAT:2, Tag:4,  
              LenBits:2, Body/binary>>, Context) ->  
{PacketData, S2Rest} = case LenBits of  
  0 -> <<Length, Obj:Length/binary, SRest/binary>> = Body, {Obj, SRest};  
  1 -> <<Length:16, Obj:Length/binary, SRest/binary>> = Body, {Obj, SRest};  
  2 -> <<Length:32, Obj:Length/binary, SRest/binary>> = Body, {Obj, SRest}  
end,  
NewContext = decode_packet(Tag, PacketData, Context),  
decode_packets(S2Rest, NewContext).
```

Alternatíva: <http://kaitai.io/> vs.

https://github.com/kaitai-io/kaitai_struct/issues/27

	Burp Extender	mitmproxy	Piper
Programming language(s)	JVM: Java, Kotlin, Jython, JRuby ...	Python 3	∇
Development cycle	slow	fast	fast
Composability	low	low	high

This matters, since **pentesting is all about improvization and one-off solutions.**

Source code and binaries under GPL:
<https://github.com/silentsignal/burp-piper>
(also in BApp Store)

Language of choice (Piper)



- ▶ Jython: worst of both worlds
- ▶ JRuby: same as Jython
- ▶ Java: I still don't like it
- ▶ Kotlin: better syntax at least
- ▶ <https://kotlinlang.org/docs/reference/comparison-to-java.html>

Comparison: Python + git diff



```
diff --git a/tmp/piper-8350675374160785262.bin b/tmp/piper-613289896204064887.bin
index a2def11..667609c 100644
--- a/tmp/piper-8350675374160785262.bin
+++ b/tmp/piper-613289896204064887.bin
@@ -2047,7 +2047,7 @@
     "disabled": false,
     "downloads_url": "https://api.github.com/repos/dnet/chrl/downloads",
     "events_url": "https://api.github.com/repos/dnet/chrl/events",
-    "fork": false,
+    "fork": true,
     "forks": 0,
     "forks_count": 0,
     "forks_url": "https://api.github.com/repos/dnet/chrl/forks",
@@ -2268,7 +2268,7 @@
     "node_id": "MDEwOlJlcG9zaXRvcnk0OTE2MDA0",
     "notifications_url": "https://api.github.com/repos/dnet/cmp/notifications?since=all,participating",
     "open_issues": 0,
-    "open_issues_count": 0,
+    "open_issues_count": 1,
     "owner": {
       "avatar_url": "https://avatars1.githubusercontent.com/u/163115?v=4",
       "events_url": "https://api.github.com/users/dnet/events{/privacy}",
```

Comparison: Burp binary diff



51	74 73 65 6e 63 72 79 70	74 2e 6f 72 67 30 82 01	tsencrypt.or...
52	05 06 0a 2b 06 01 04 01	d6 79 02 04 02 04 81 f6	..+&<0y...
53	04 81 f3 00 f1 00 77 00	6f 53 76 ac 31 f0 31 19	≥óñwoSv-1...
54	d8 99 00 a4 51 15 ff 77	15 1c 11 d9 02 c1 00 29	0αQ&ywo>D...
55	06 8d b2 08 9a 37 d9 13	00 00 01 6b db 38 0c c5	α²/7Uj;<kÜ...
56	00 00 04 03 00 48 30 46	02 21 00 91 c9 84 48 b0	≥>H0F!ÉH°
57	c5 1f b4 cd 48 ca af b4	0e 48 cd 9d 27 b7 8a 6c	Åö'íHÉ'..Hí'í
58	44 6c 9b cc 70 30 09 6c	68 7a a4 02 21 00 b5 78	Dlíp0lhza!µx
59	83 ea 4b c4 32 21 ab 85	51 be 9b 59 bc 5d df 3f	éKÄ2!«Q%Y...
5a	8f b6 ca 3c d6 ba 4b 8e	45 5c fb 7d ca 22 00 76	¶É<0&KEVÜ]...
5b	00 63 f2 db cd e8 3b cc	2c cf 0b 72 84 27 57 6b	còÜfè;í;írwk
5c	33 a4 8d 61 77 8f bd 75	a6 38 b1 c7 68 54 4b d8	3αaw½u;8±...
5d	8d 00 00 01 6b db 38 0c	c3 00 00 04 03 00 47 30	«kÜ8-Å¿>G0
5e	45 02 21 00 fc 1d f4 ee	16 60 d0 2b 24 ad 52 af	EÜü öf&°D+...
5f	fc 0c 43 7d 67 29 08 ac	ed 40 00 47 de 08 77 e3	ü Cj-g) /-í@...
60	6a 6a ae 6d 02 20 32 9a	53 62 0e 7a 17 55 93 bd	jj@m 2Sb.z'...
61	ab c7 07 e7 3a 97 7c 24	96 77 76 b1 38 d1 80 1c	«Ç&ç; \$wv±...
62	07 9c 6c 19 39 7d 30 0d	06 09 2a 86 48 86 f7 0d	α;9}0α*H+
63	01 01 0b 05 00 03 82 01	01 00 6e ee bb 29 65 e8	<<<>><<ñ!>...
64	c9 3f 60 0e 6a 3c 00 19	21 47 f4 28 72 8f f6 c2	É? 'j-<¡Gô(r...
65	33 cf ce 39 3a 11 0f e1	b0 61 81 7d c0 31 97 56	3í9;»-á°a'j...

Key: Modified Deleted Added

✓ Sync views

Comparison: OpenSSL + git diff



```
diff --git a/tmp/piper-4143288618703797045.bin b/tmp/piper-2900910360535094730.bin
index bb8f3f7..e30b492 100644
--- a/tmp/piper-4143288618703797045.bin
+++ b/tmp/piper-2900910360535094730.bin
@@ -2,7 +2,7 @@
   4:d=1 hl=4 l=1364 cons: SEQUENCE
   8:d=2 hl=2 l= 3 cons: cont [ 0 ]
  10:d=3 hl=2 l= 1 prim: INTEGER :02
- 13:d=2 hl=2 l= 18 prim: INTEGER :0373C93BC585F979FF76E44641B36FB3E715
+ 13:d=2 hl=2 l= 18 prim: INTEGER :03326FAB4A09C01D141A7ED1331A8EFC826
  33:d=2 hl=2 l= 13 cons: SEQUENCE
  35:d=3 hl=2 l= 9 prim: OBJECT :sha256WithRSAEncryption
  46:d=3 hl=2 l= 0 prim: NULL
@@ -20,13 +20,13 @@
  91:d=5 hl=2 l= 3 prim: OBJECT :commonName
  96:d=5 hl=2 l= 26 prim: PRINTABLESTRING :Let's Encrypt Authority X3
 124:d=2 hl=2 l= 30 cons: SEQUENCE
- 126:d=3 hl=2 l= 13 prim: UTCTIME :190710083011Z
- 141:d=3 hl=2 l= 13 prim: UTCTIME :191008083011Z
+ 126:d=3 hl=2 l= 13 prim: UTCTIME :190710083040Z
+ 141:d=3 hl=2 l= 13 prim: UTCTIME :191008083040Z
 156:d=2 hl=2 l= 28 cons: SEQUENCE
 158:d=3 hl=2 l= 26 cons: SET
 160:d=4 hl=2 l= 24 cons: SEQUENCE
 162:d=5 hl=2 l= 3 prim: OBJECT :commonName
- 167:d=5 hl=2 l= 17 prim: UTF8STRING :*.silentsignal.eu
+ 167:d=5 hl=2 l= 17 prim: UTF8STRING :*.silentsignal.hu
 186:d=2 hl=4 l= 546 cons: SEQUENCE
 190:d=3 hl=2 l= 13 cons: SEQUENCE
 192:d=4 hl=2 l= 9 prim: OBJECT :rsaEncryption
@@ -56,13 +56,13 @@
 881:d=5 hl=2 l= 99 prim: OCTET STRING [HEX DUMP]:3061302E06082B06010506073001B622687474703A2F2F6F6373702E696E742078332E6C6574
 982:d=4 hl=2 l= 45 cons: SEQUENCE
 984:d=5 hl=2 l= 3 prim: OBJECT :x509v3 Subject Alternative Name
- 989:d=5 hl=2 l= 38 prim: OCTET STRING [HEX DUMP]:302462112A2E73696C656E747369676E616C2E65756820F73696C656E747369676E616C2E6575
+ 989:d=5 hl=2 l= 38 prim: OCTET STRING [HEX DUMP]:302462112A2E73696C656E747369676E616C2E68756820F73696C656E747369676E616C2E6875
 1029:d=4 hl=2 l= 76 cons: SEQUENCE
```

- ▶ bug bounty: nesze itt egy hiba
- ▶ teljes körű vizsgálat: bizonyít(ani próbál)om, hogy a rendszer biztonságos
- ▶ screenshot: egyszerű készíteni, az alapján újracsinálni nehéz
- ▶ „futtatható dokumentáció”: nagyobb befektetés, cserébe reprodukálható

- ▶ modern fejlesztő/üzemeltető: fúj, nem típusos, nehezen karbantartható
- ▶ REPL before it was cool, interaktív + script egyben
- ▶ pipe, subshell: kis blokkokból összelegőzünk komplex megvalósítást
- ▶ fejből pár gyakori hasznosság:
 - ▶ tr – karakterhelyettesítés (vö. URL-safe base64), -d kapcsolóval törlés (vö. újsorok törlése)
 - ▶ cut – *n*. oszlop kiválasztása karakterpozíció vagy határolójel alapján
 - ▶ grep – reguláris kifejezésre szűrés (hasznos: -o)
 - ▶ sed – reguláris kifejezés kiértékelése
 - ▶ awk – all of the above ;)

- ▶ <https://rada.re/>
- ▶ Unix filozófia, külön is hasznos toolok
- ▶ rax2 – konverzió számrendszerek között + bájtokra
 - ▶ `rax2 -s 414141 ⇒ AAA`
 - ▶ `... | rax2 -S ⇒ hexpairs`
 - ▶ utóbbi kettő base64 `-w0` paranccsal kombinálva kiváló
 - ▶ `rax2 0x41 ⇒ 65`
- ▶ `radiff2` – bináris diff, akár disassemblerrel is
- ▶ `rahash2` – hash akár egy fájl egy részéről is

- ▶ lib és CLI
- ▶ konverzió: x509, pkcs12, rsa, ec, pkey
- ▶ titkosítás: rsautl, dgst (-sign és -verify), enc
- ▶ kulcsgenerálás: genpkey (új), genrsa (rég)
- ▶ s_client – SSL/TLS kliens
- ▶ req – CSR ill. self-signed tanúsítvány műveletek
- ▶ asn1parse – ASN.1 DER dekódolás, javasolt kapcsolók: -i, -strparse

- ▶ <https://frida.re/>
- ▶ dinamikus instrumentáció: reprodukálható debugger-helyettesítő
- ▶ gadget: másik OS, másik arch: <https://github.com/frida/frida/releases/>
- ▶ trace: gyors képet ad érdekes függvényekről, beavatkozást is lehetővé tesz
 - ▶ -a 'Core!0x105ae0' – ha csak a cím van meg
 - ▶ -i '*rypt*' – minden ilyen nevű szimbólumra ugrik
 - ▶ -m '+[Curve25519 *:]' – Objective-C iOS-en
- ▶ objection: iOS és Android app hacking – <https://github.com/sensepost/objection>
- ▶ TLS kulcsszerzésre lásd korábbi előadásom
 - ▶ <https://vimeo.com/425497704>
 - ▶ https://silentsignal.hu/docs/S2_VSzA_MITM_2020.pdf

- ▶ bat – syntax highlight: <https://github.com/sharkdp/bat>
- ▶ JSON
 - ▶ python -m json.tool – built-in JSON pretty-printing
 - ▶ gron – grepelhető JSON: <https://github.com/tomnomnom/gron>
 - ▶ jq – XPath-szerűség JSON-re
- ▶ XML
 - ▶ xmllint --format – XML pretty-printing
 - ▶ xmllint --xpath – XPath

- ▶ iptables – célpont csomagok gyors megszerzése
- ▶ tcpsocks – furcsa csomagok SOCKS-ba irányítása: <https://github.com/vi/tcpsocks>
- ▶ Wireshark + dissectoraik ⇒ exportok
 - ▶ JSON – szeret dupla kulcsokat csinálni
 - ▶ PDML – XML, de legalább well-formed
- ▶ Scapy – tetszőleges csomag összerakása és válasz feldolgozása Pythonból
<https://scapy.net/>

Példák: Wi-Fi



Melyik csatornákon volt AP?

```
$ cut -f 6 -d ';' r.kismet.csv | sort -un
```

Channel

```
1  
6  
11  
36  
40  
44  
48
```

```
$ xmllint --xpath '//wireless-network/channel/text()' r.kismet.netxml | sort -un
```

Példák: apktool



Smali bájtkód módosítása után fordítás + aláírás + újrategelés – opcionálisan bővíthető logcat futtatással, ami meg gregelhető...

```
apktool b --use-aapt2 && (cd dist ; jarsigner -sigalg SHA1withRSA -digestalg SHA1  
-keystore ../objection/utis/assets/objection.jks -storepass basil-joule-bug  
*.apk objection && adb install -r *.apk)
```

Példák: cert pinning konzisztencia



Network Security Config XML-ben felsorolt hash-ek szerepelnek a bájtkódban? Ha igen, hol?
(Tipikus példa: OkHttp pinning)

```
$ xmllint --xpath '//pin/text()' res/xml/network_security_config.xml  
  | tr '=' '\n' | sort -u | while read i; do echo $i; grep -Frl $i; done
```

Példák: Android KeyStore feloldás



10152_USRPKEY_pinKey fájlban lévő AES-256 kulccsal oldható fel a megadott Base64 kódolással Shared Preferencesbe mentett kriptogram (IV mellette ugyanúgy ugyanott)

```
$ base64 -d <<<746nQ5wQYejBNx8Q5Jt4iw== | openssl aes-256-cbc -d  
-K $(dd if=10152_USRPKEY_pinKey bs=1 skip=45 count=32 2>/dev/null | rax2 -S)  
-iv $(base64 -d <<<3tfqSUrQaSvbeH9UvNU5dg== | rax2 -S)
```

Példák: paraszt XML parsing



Nem volt kéznél xmllint, és a végén még protobuf dekódolás is kellett:

```
$ grep string tmp/main_preferences.xml | cut -f 2 -d '>' | cut -f 1 -d '<'  
  | head -n1 | rax2 -s  
  | protoc --decode=google.crypto.tink.EncryptedKeyset tink.proto
```

Példák: iOS jailbreak detekció



Elegáns minimális változtatás:

```
$ radiff2 -a arm -b 64 -D CensoredID*  
--- 0x00129bd0 e00318aa  
- mov x0, x24  
+++ 0x00129bd0 000080d2  
+ movz x0, 0
```


- ▶ más igények más megoldások
- ▶ ha még nem ijesztettünk el, keresünk:
 - ▶ junior pentestert
 - ▶ senior pentestert
 - ▶ C# fejlesztőt

KÖSZÖNÖM!

VERES-SZENTKIRÁLYI ANDRÁS

vsza@silentsignal.hu



facebook.com/silentsignal.hu



@SilentSignalHU



@dn3t

