

# Knocking down the HACIENDA with TCP Stealth

Christian Grothoff  
Actual work: Julian Kirsch



*inria*  
informatiques mathématiques



Technische Universität München

July 23, 2015

# What is HACIENDA?

---

- Data reconnaissance tool developed by the CITD team in JTRIG
- Port Scans entire countries
  - Uses nmap as port scanning tool
  - Uses GEOFUSION for IP Geolocation
  - Randomly scans every IP identified for that country



**NAC**  
NETWORK ANALYSIS CENTRE

UK TOP SECRET STRAP1  
TOP SECRET//COMINT//REL FVEY



# How is it used?

---

- CNE
  - ORB Detection
  - Vulnerability Assessments
- SD
  - Network Analysis
  - Target Discovery



## Step 3

# Hacking in SIGINT



# The Hacking Process

1. (R)econnaissance
2. (I)nfection
3. (C)ommand And Control
4. (E)xfiltration

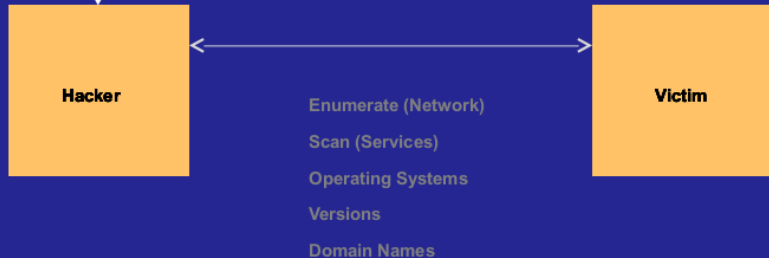


# Reconnaissance

## Publicly Available Information

(Email Address, Location, Network Info, Passwords, etc.)

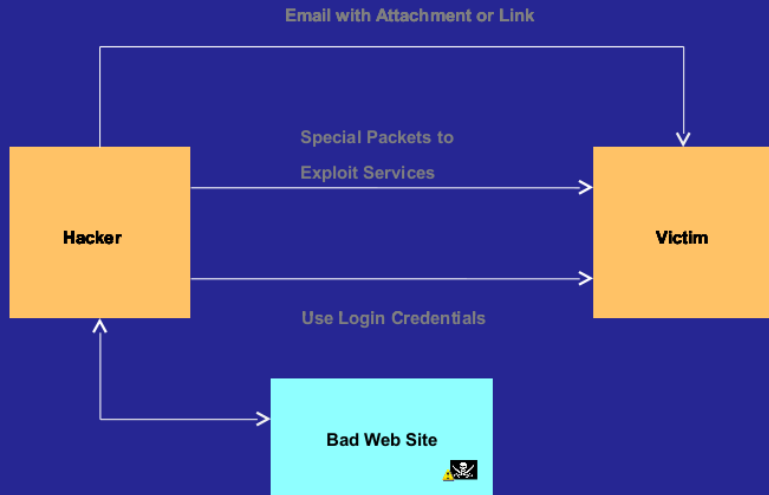
Research



Reconnaissance Infection Command and Control Exfiltration



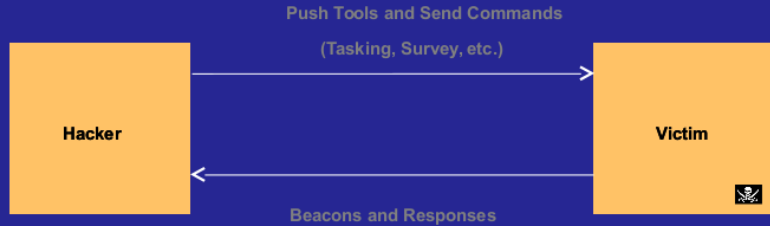
# Infection



Reconnaissance Infection Command and Control Exfiltration



# Command and Control



Reconnaissance Infection **Command and Control** Exfiltration





# Exfiltration

Exfil using known and custom protocols  
(Known: HTTP, SMTP, ICMP, FTP, etc)



# How is it used?

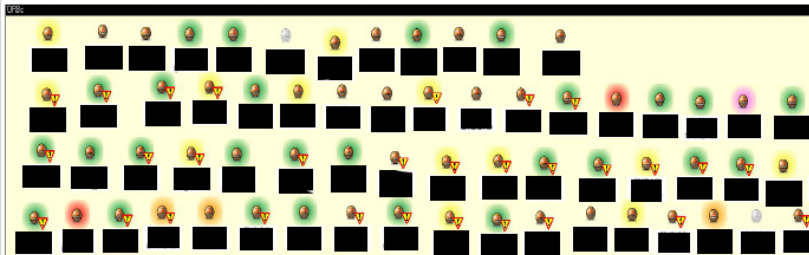
---

- CNE
  - ORB Detection
  - Vulnerability Assessments
- SD
  - Network Analysis
  - Target Discovery

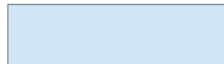


# LANDMARK

- CSEC's Operational Relay Box (ORB) covert infrastructure used to provide an additional level of non-attribution; subsequently used for exploits and exfiltration
- 2-3 times/year, 1 day focused effort to acquire as many new ORBs as possible in as many non 5-Eyes countries as possible







GSM provider

- \* NSA TAO requested assistance gaining access to the network
- \* Network analysis using OLYMPIA:
  - \* DNS query to determine IP address
  - \* IP address to network range
  - \* Network range to port scan
  - \* Are there any vulnerable devices in that range?
- \* Duration: < 5 minutes

So, is it all lost?

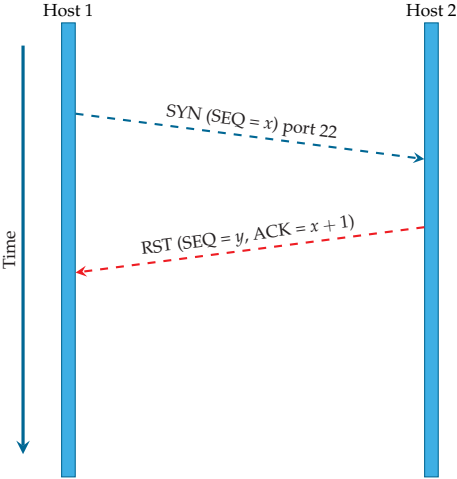


# Two Solutions

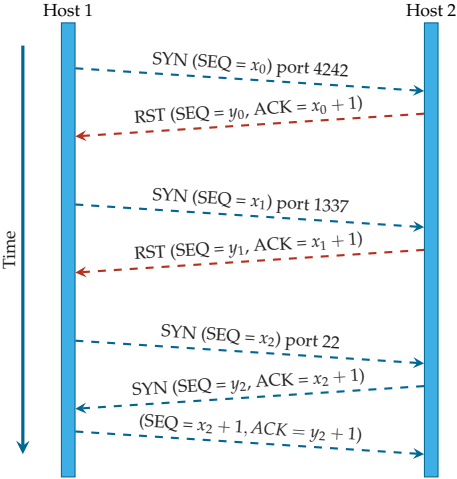
- ▶ Backwards-compatible minimally invasive hotfix (TCP Stealth)
- ▶ Clean-slate principled rearchitecture

# An Introduction to Port Knocking

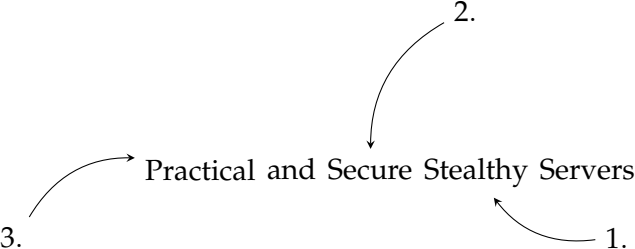
### No knock, no fun

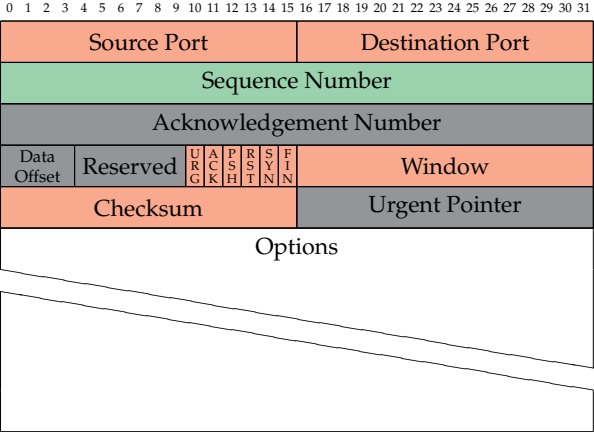


### Port knocking example









# Design (v1)

## Security

- ▶ Destination IP address  $IP_d$
- ▶ Destination port  $P_d$
- ▶ TCP timestamp  $T$
- ▶ Pre-Shared Key  $S$
- ▶ Hash function  $h$

### Authentication Security Token (AV)

$$AV := h((IP_d, P_d, T), S)$$

- ▶ ISN := AV

# SECONDDATE

- SECONDDATE is an exploitation technique that takes advantage of web-based protocols and man-in-the-middle (MitM) positioning.
- SECONDDATE influences real-time communications between client and server and can quietly redirect web-browsers to FA servers for individual client exploitation.
- This allows mass exploitation potential for clients passing through network choke points, but is configurable to provide surgical target selection as well.

# Design (v2)

## Security

- ▶ Destination IP address  $IP_d$
- ▶ Destination port  $P_d$
- ▶ TCP timestamp  $T$
- ▶ Pre-Shared Key  $S$
- ▶ Hash functions  $h, h'$
- ▶ Payload  $p$

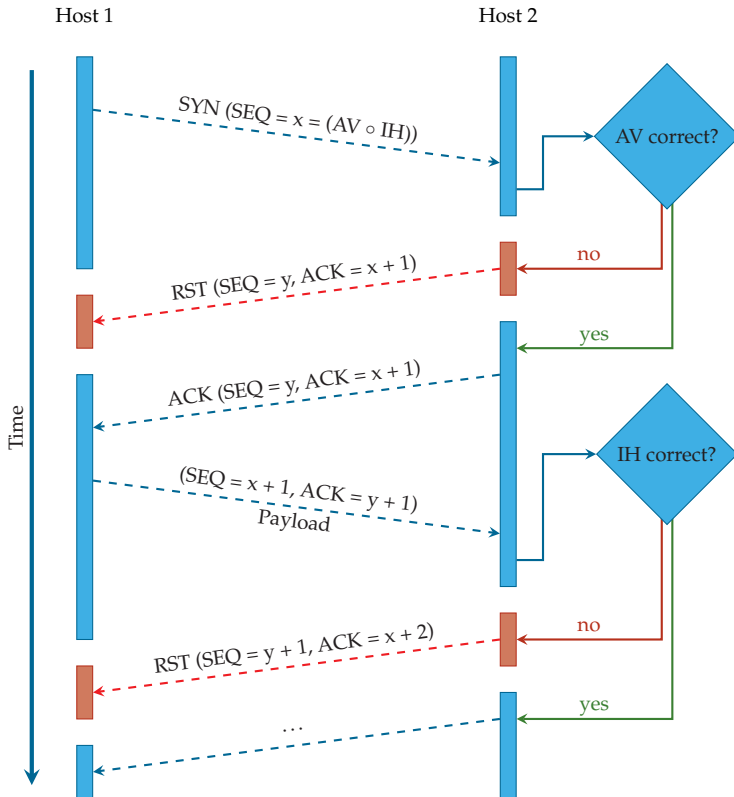
### TCP Payload Integrity Protector IH

$$IH := h'(S \circ p)$$

### Authentication Security Token AV

$$AV := h((IP_d, P_d, T, IH), S)$$

- ▶  $ISN := AV \circ IH$



# Design

## Ease of Use

- ▶ Source IP and Port *not* included in ISN generation
  - ⇒ Compatibility with NATs
- ▶ Knocking is implemented *in the kernel*
  - ⇒ No fiddling with config-files, firewall rules or daemons
  - ⇒ Trivial to use from an application developer's perspective

# Design

## Ease of Use – TCP Stealth Server

```
1 char secret[64] = "This is my magic ID.";
2 int payload_len = 4;
3 int sock;
4
5 sock = socket(AF_INET, SOCK_STREAM, IPPROTO_TCP);
6 if (sock < 0) {
7     printf("socket() failed, %s\n", strerror(errno));
8     return 1;
9 }
10 if (setsockopt(sock, IPPROTO_TCP, TCP_STEALTH, secret, sizeof(secret)) {
11     printf("setsockopt() failed, %s\n", strerror(errno));
12     return 1;
13 }
14 if (setsockopt(sock, IPPROTO_TCP, TCP_STEALTH_INTEGRITY_LEN,
15     &payload_len, sizeof(payload_len))) {
16     printf("setsockopt() failed, %s\n", strerror(errno));
17     return 1;
18 }
19 /* Continue with bind(), listen(), accept(), recv(), ... */
```



# Design

## Ease of Use – TCP Stealth Client

```
1 char secret[64] = "This is my magic ID.";
2 char payload[4] = "1234";
3 int sock;
4
5 sock = socket(AF_INET, SOCK_STREAM, IPPROTO_TCP);
6 if (sock < 0) {
7     printf("socket() failed, %s\n", strerror(errno));
8     return 1;
9 }
10 if (setsockopt(sock, IPPROTO_TCP, TCP_STEALTH, secret, sizeof(secret)) {
11     printf("setsockopt() failed, %s\n", strerror(errno));
12     return 1;
13 }
14 if (setsockopt(sock, IPPROTO_TCP, TCP_STEALTH_INTEGRITY,
15     payload, sizeof(payload)) {
16     printf("setsockopt() failed, %s\n", strerror(errno));
17     return 1;
18 }
19 /* Continue with connect(), send(), ... */
```

# Design

## Ease of Use – libknockify

- ▶ Shared library for use at compile- or run-time
- ▶ Enables TCP Stealth functionality for legacy code

```
$ LD_PRELOAD=./libknockify.so ncat knock-server application-port
```

- ▶ Configuration options (such as the TCP Stealth secret) are given as environment variables or via a special file

# Limitations

- ▶ Distribution of the Pre-Shared Key
- ▶ ISN has only 32 bits

# Limitations

- ▶ Distribution of the Pre-Shared Key
- ▶ ISN has only 32 bits
- ▶ Changes to ISN and TSVal by middle boxes:

Behavior	TCP Port		
	34343	80	443
Unchanged	<b>126 (93%)</b>	116 (82%)	128 (90%)
Mod. outbound	5 (4%)	5 (4%)	6 (4%)
Mod. inbound	0 (0%)	1 (1%)	1 (1%)
Mod. both	4 (3%)	13 (9%)	7 (5%)
Proxy (probably mod. both)	0 (0%)	7 (5%)	0 (0%)
Total	135 (100%)	142 (100%)	142 (100%)

Numbers by Honda et al. "Is it Still Possible to Extend TCP?"

## Working Code...

- ▶ Implemented for 3+ Linux kernel versions
- ▶ Implemented for FreeBSD
- ▶ Holger Kenn (MSFT) said would be easy to do in W32-Kernel(s)
- ▶ Sample client and server programs
- ▶ Patches for OpenSSH, GNUnet, systemd
- ▶ libknockify(.so) LD\_PRELOAD
- ▶ Master's thesis, presentations, website, article in 5 languages
- ▶ Tested in big-endian/little-endian platforms (incl. compatibility)
- ▶ Draft has test vectors, detailed protocol specification
- ▶ Based on 1 year of community feedback, authors clueless about what else to do. Except find the right WG. Spencer solved that.

## Why standardize...

- ▶ Port scanning is a well-known vulnerability. We need to address it.
- ▶ Implementations need to be compatible.
- ▶ Kernels must offer it for ease of deployment.
- ▶ Kernels will only ship by default if standardized.
- ▶ (Some GNU/Linux distributions already ship this anyway.)
- ▶ This does not solve all issues, but as many as we can with maximum backwards compatibility.

## ... and rough consensus?

### Find more information at:

- ▶ <https://gnunet.org/>
- ▶ <https://gnunet.org/knock>
- ▶ <https://gnunet.org/gns>
- ▶ <https://gnunet.org/mcb>

### Thanks to:

JULIAN KIRSCH  
JACOB APPELBAUM  
MONIKA ERMERT  
LAURA POITRAS  
HENRIK MOLTKE  
MAURICE LECLAIRE  
ANDREAS ENGE  
BART POLOT  
LUCA SAIU  
THE SOURCE

This work was funded  
by the Deutsche  
Forschungsgemein-  
schaft (DFG) under  
ENP GR 3688/1-1.

Slides will be at <http://grothoff.org/christian/>.