



WAVESTONE

Red Teamer 2.0

Automating the C&C Set up Process

Charles IBRAHIM
December 7th, 2018



Before anything, thanks !

- ✓ If live streaming goes to heaven, this is for you, ya **Baba**
- ✓ My **wife** is wonderful
- ✓ The **Botconf reviewing committee** has been generous
- ✓ My boss is cool: thx **Yann Filliat**
- ✓ Thx to **Wavestone's audit pool** & his R&D lead **Arnaud Soullié**
- ✓ Thanks to **Nicolas Mattiocco** - @MaKyOtOx for his inputs and for Patrowl whose structure was helpful to me to get hands dirty on Django
- ✓ Thanks to **Cyril Mansour, Ahmed Fendri & Mohamed Reda Feknous**, who've contributed to the project

Who am I ?

Information Security Consultant, dad, husband, lot of sport and miniatures painting.

Main experiences

Pentests – Numerous clients

Setting a Command and Control architecture – Transversal

Head of SOC & CERT Caisse des Dépôts – Informatique Caisse des Dépôts

Specialties

- / Technical audits and architecture evaluation
- / Cyberattacks detection methods and techniques
- / Incident response

Publications

- / MISC issues: september/october 2018 & march/april 2018 & september/october 2017.
- / Speaker at ESIEA SECURE EDITION 2016, at [TF-CSIRT 49th meeting in Zürich](#), at [FIC 2017](#), at [TF-CSIRT 51st meeting in The Hague](#), at Splunk Live 2017, at EPITA & École 42.

Charles IBRAHIM

Senior consultant

> 5 years of experience



I did not find a funny meme
sorry

Wavestone auditors – who are we ?

 @secuinsider



Developers

- / **PyKEK**: Kerberos exploitation framework. **First MS14-068 public exploit**
- / **Metasploit doubles**: modbusclient & modicon_stux_transfer
- / **Script scan7**: Siemens PLC dialog interface *via* S7
- / **Burp extensions**: Using .pac proxy files, Java deserialization, etc.



Writers

- / **MISC**
 - > N°99 : *Stealthy communication techniques with **Empire***
 - > N°96 : **PowerView** or how to become domain Admin faster
 - > N°82 : Introduction to **Burp** extensions development
 - > N° 77 : « Let's hook » with **JavaSnoop!**
 - > N°74 : Intrusion tests on industrial PLC

/ SecurityInsider blog



- / **Wavestone top 10 web vulnerabilities**



Talkers

- / **Red Teamer 2.0: Automating the C&C Set up Process** : Botconf 2018
- / **Pentesting Active Directory**: Bsidés Lisbon 2018
- / **Hadoop Safari**: Zeronights 2016, PHDays 2017, BSides Las Vegas 2017, HITB Singapour 2017
- / **Transactions on z/OS CICS**: Zeronights 2016, Hitcon 2016, HITB Amsterdam 2017
- / **Industrial Control Systems: pentesting PLCs 101**: Brucon 2017, Bsidés Las Vegas 2015, BlackHat Europe 2014
- / **Is it possible to secure a Windows domain ?** JSSI 2014



Enthusiasts

- / **OSSIR**: OSSIR Paris group co-facilitators
- / **GreHack**: Gold Sponsor



- / **SIGSEGV1 (RTFM)** : Gold Sponsor



... and teachers

Courses, seminars about Information Security
... and trainings tackling **ICS, Mobile, or Web** intrusion tests



confidential | © WAVESTONE



/ **01**

Why, on earth, another tool ?

Reminder: What is a red team operation ?



Test an information system toughness deeply and in real conditions:

- / Miscellaneous attack vectors
- / Bounces on indirect targets to reach the goal



Physical intrusions

On-site physical access to facilitate the target approach or bypass security access measures



Logical intrusions

Web intrusions, sinkholes, keyloggers, ...



Social engineering

Trapping individuals to gain access to sensitive informations, escalate privileges, ...

We use the latest exploits... or try developing them 😊

We go further than during classical audits

... but we keep control at all time

Introduction

Why, on earth, another tool ?

Because



- / Tons of tools
- / No aggregation
- / OPSEC fails easy



What do we want ?

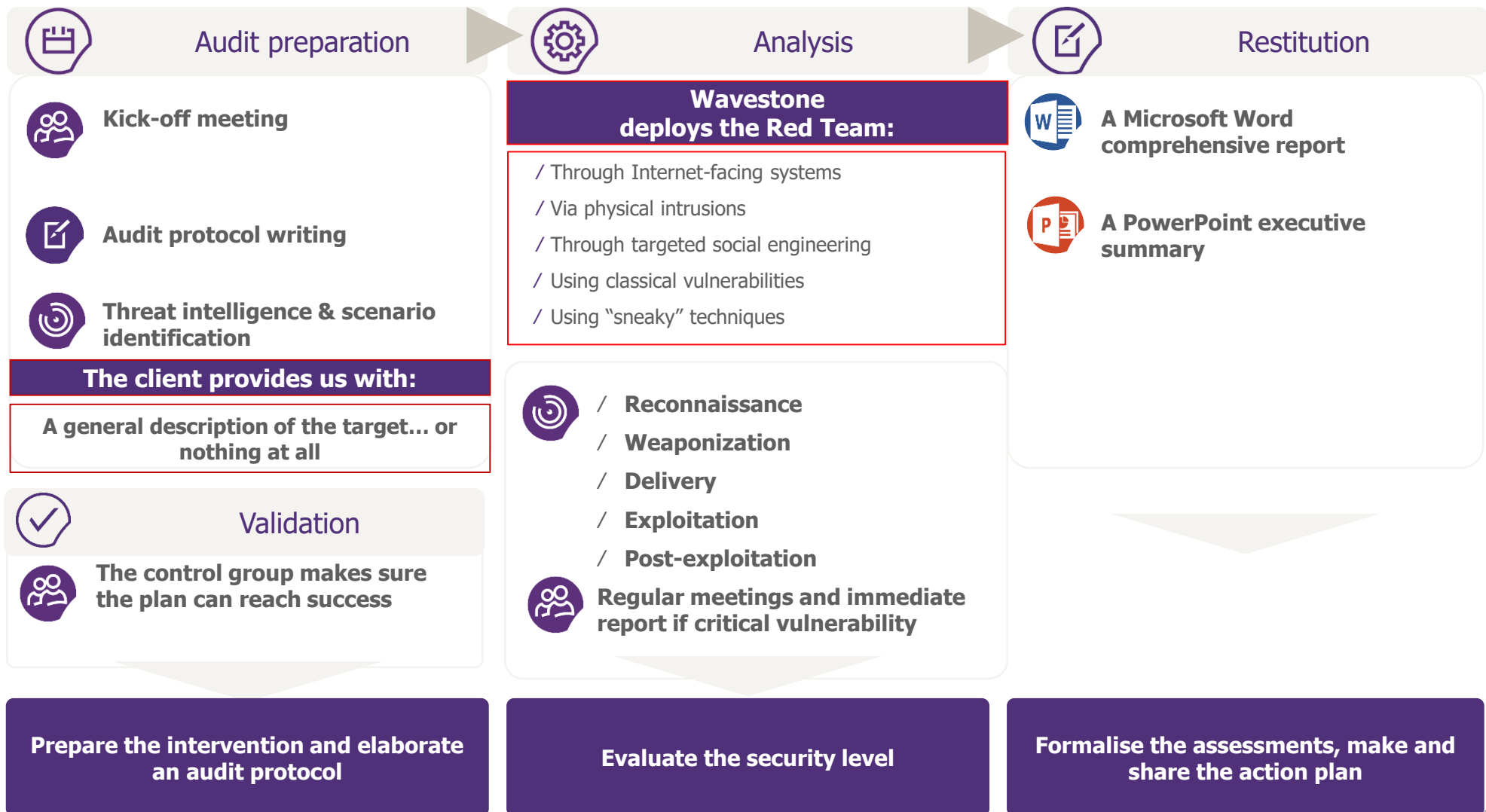
- / Facilitate Red Team operations by:
 - Reducing the time-to-build-an-infrastructure
 - Easing common actions launching
 - Enabling complex actions with 1 or 2 clicks
 - Enabling a long operation organisation & reporting
 - Reducing the OPSEC fails risk



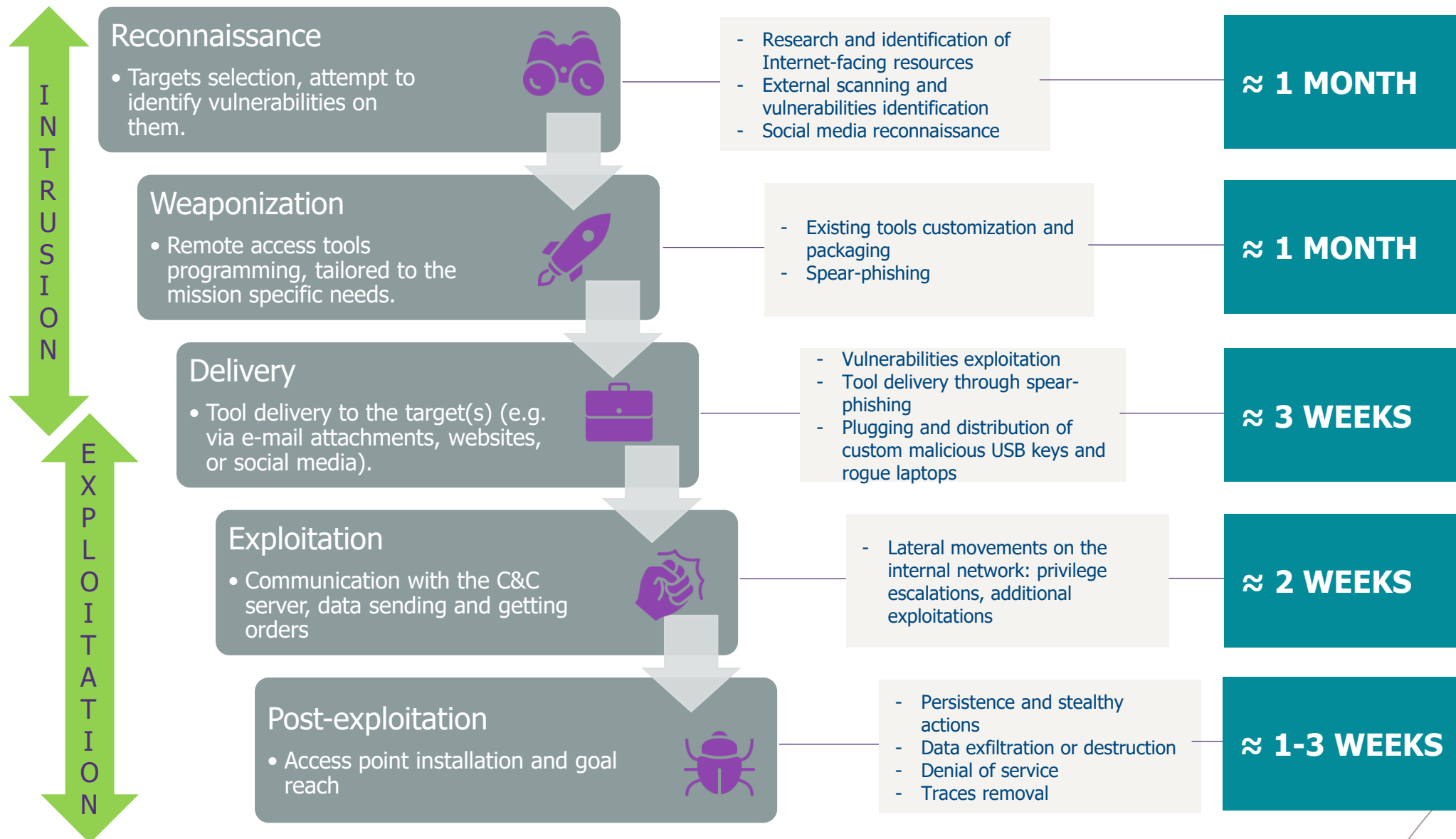
/ **02**

Our approach – what is this tool about?

Reminder: the approach in leading a red team operation



How long would it take to professional attackers to perform...



Methodological views

Red team exercises are not audits

It might be hard to make the business accept associated risks

What works well (generally)

- / **Communication** with the control group (**frequent**)
- / **Discussions** with the control group leading to identify **relevant attack scenarios**
- / **Variety of intrusion types:** physical & logical, remote & on-site, using classical penetration testing techniques as well as more real-life inspired, custom methods

Possible issues

- / **Keeping the control group number small**
- / **Separating Red and Blue teams** in a more clear-cut way
- / **Defining roles and responsibilities** of each stakeholder prior to the instance, and **stick to them**
- / **Switching from a “stop-and-go”** model (where red teamers must ask for clearance at each step) **to an “emergency stop” model** (where the client is kept informed of each next step, and can stop it if required)

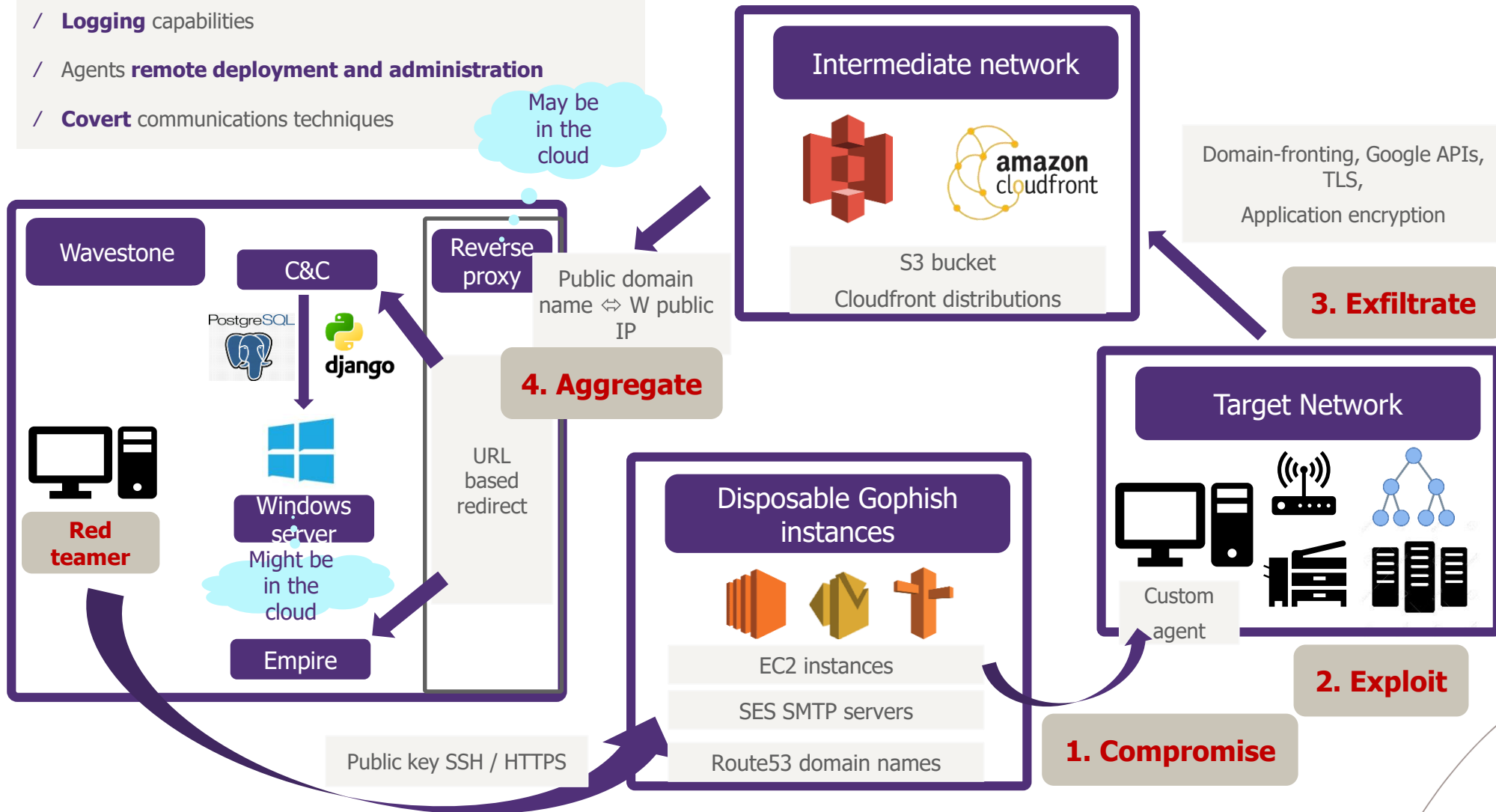


/ **03**

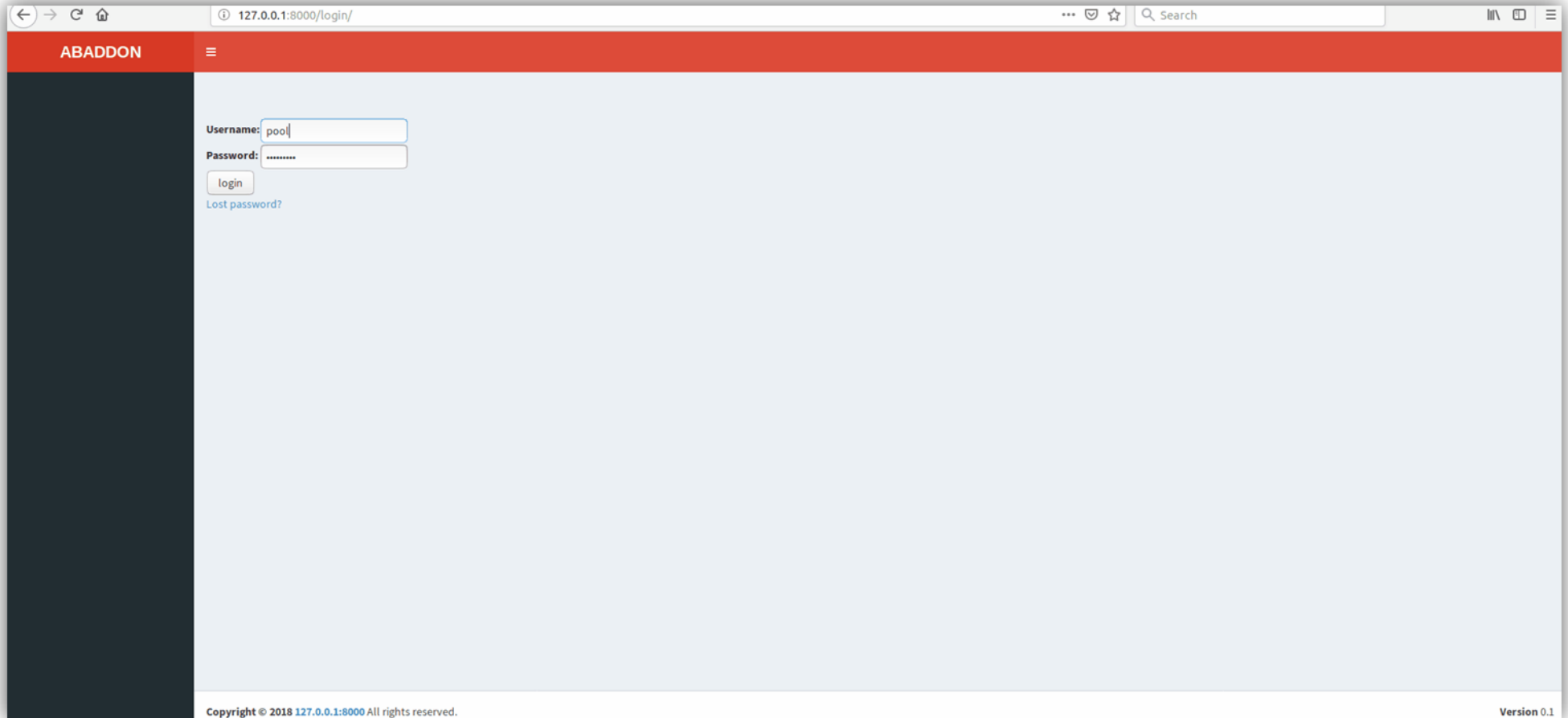
Detailed views

Command and Control architecture

- / Users **authentication** (done) & **data segregation** (to be implemented) between red teamers
- / **Logging** capabilities
- / Agents **remote deployment and administration**
- / **Covert** communications techniques



Show me that tool !



What works now

/ The application works and is useful

What could be improved

- / Tons of things (which that presentation is about), and among them:
 - RT operation as a project management (dashboard, alerts, multiple users collaboration, etc.)

A crucial step that could be enhanced



Goal in that phase: Characterize as accurately as possible the targets, remotely or on location.



What we've done: Passive reconnaissance automation

What we've automated

/ From a custom external cartography methodology, we took the recon-ng part and put it to an interface

What could be improved

/ More recon-ng lookups

/ Automate discover.sh & domain tools API & Shodan lookups

/ Social engineering, physical intrusion: what do you think ?

Show me your reconnaissance !



What does it look like, really ?

The screenshot displays the ABADDON web application interface. On the left, a terminal window shows logs for a Django development server. The main interface features a sidebar with a user profile 'pool' and a 'MAIN NAVIGATION' menu with options: Dashboard, Reconnaissance, Weaponization, Delivery, Exploitation, and Post-Exploitation. The central area is titled 'Infrastructure dashboard' and includes a 'Domain:' input field, a 'Favorite modules:' section with checkboxes for google_site_web, bing_domain_web, netcraft, hackertarget, and brute_hosts, and a 'Run Recon-ng' button. The footer contains the text 'Copyright © 2018. All rights reserved.' and 'Version 0.1'.

Compile a lightly obfuscated payload on-the-fly



Goal: Develop an actionable & stealthy exploitation tool



What we've done: An interface (& a web service) to compile a custom Remote Access Tool (RAT)

What we've automated

- / Compile a **RAT that (currently) does not trigger** any antivirus alarms on the target network & can communicate through domain-fronting
- / ... and enables to launch an obfuscated Empire agent from (*id est* inside a Popen created by) the RAT 😊
- / **Specify in an html form** the compilation server, domain-fronting enabled cloudfront distribution, the visible-by-the-blue-team domain, and a persistence mechanism presence

What could be improved

- / Do the same with tailor-made USB keys and keyboards
- / Avoid that the domain name bought for hosting the payload be **detected as a typo squatting name**:
https://static.sstic.org/rumps2018/SSTIC_2018-06-14_P10_RUMPS_13.mp4
- / Use .NET calls instead of powershell for advanced post-exploitation actions

Parenthesis - What did you say ? Physical intrusions ?



Would you use that keyboard ?



Sent keyboards



Modified USB receiver



Dropped USB key

Show me your compilation !



What does it look like, really ?

The screenshot displays two side-by-side windows from a Windows desktop environment.

The left window is a Mozilla Firefox browser showing the 'ABADDON' web application. The address bar indicates the URL '127.0.0.1:8000/exploitation/compile'. The page features a red header with the 'ABADDON' logo and a navigation sidebar on the left. The main content area is titled 'Compile a RAT' and contains a form with the following fields:

- Compilation server:** 192.168.56.101:8000
- Cloudfront distribution:** d375443jghd.cloudfront.net
- Face domain:** cdn.example.net
- Persistence:** Yes

A blue 'Compile' button is located at the bottom of the form. The footer of the application shows 'Copyright © 2018. All rights reserved.' and 'Version 0.1'.

The right window is a Windows PowerShell terminal titled 'Select Administrator: Windows PowerShell'. The command prompt shows the current directory as 'C:\Users\IEUser\Desktop\Abaddon\compile_server>'. The terminal background is dark blue.

A robust architecture to send phishing mails



Goal: deliver the “weapon”



What we’ve done: automatically deploy Gophish on a disposable EC2 instance, configure Gophish to use SES smtp servers

What we’ve automated

- / **EC2 instance creation** (including instance profile, security group, public & private key, role & IAM strategy)
- / **Gophish deployment on one or several instances**
- / Create a dedicated s3 bucket for the cloudfront distribution (see after)

What could be improved

- / Automate the Apache web server & Nginx/EC2 reverse-proxy deployment
- / Monitor their logs
- / Integrate the Gophish results retrieval in Abaddon
- / Create a display a database model for the created infrastructure
- / Test terraform: <https://youtu.be/aEIUqrFiBb8>


Show me your distribution !



What does it look like, really ?

```
usr/lib/python3/dist-packages/requests/__init__.py:80: RequestsDependencyWarning: urllib3 (1.23)
RequestsDependencyWarning)
system check identified no issues (0 silenced).
November 23, 2018 - 10:49:21
jango version 2.1.3, using settings 'abaddon.settings'
starting development server at http://127.0.0.1:8000/
uit the server with CONTROL-C.
```

ABADDON

 pool

MAIN NAVIGATION

- Dashboard
- Reconnaissance
- Weaponization
- Delivery
- Exploitation
- Post-Exploitation

Cloudfront deployment dashboard

Deployed distribution id:

Versions HTTP prises en charge

- ☒ HTTP/2, HTTP/1.1, HTTP/1.0
- ☐ HTTP/1.1, HTTP/1.0


Objet racine par défaut


Journalisation


- ☒ Activé
- ☐ Désactivé


Compartiment pour les journaux


Préfixe de journal












Show me your EC2 !









What does it look like, really ?


```
...s17/c10/python3/dist-packages/requests/..._init_.py:88: RequestsDependencyWarning: urllib3 (1.25)
RequestsDependencyWarning)
system check identified no issues (0 silenced).
November 22, 2018 - 16:28:50
Django version 2.1.3, using settings 'abaddon.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
22/Nov/2018 16:28:53] "GET / HTTP/1.1" 200 11487
Not Found: /favicon.ico
22/Nov/2018 16:28:53] "GET /favicon.ico HTTP/1.1" 404 4129
Not Found: /favicon.ico
22/Nov/2018 16:28:53] "GET /favicon.ico HTTP/1.1" 404 4129
```

ABADDON

 pool

MAIN NAVIGATION

-  Dashboard
-  Reconnaissance <
-  Weaponization <
-  Delivery <
-  Exploitation <
-  Post-Exploitation <

 pool

Home

Welcome, red team angel!

By clicking somewhere left, you will be able to:

- Perform reconnaissance actions
- Scan your target
- Set up gophish on EC2 instances
- Set up a Cloudfront distribution enabling domain-fronting (one click)
- Receive incoming connections from domain-fronting configured RATs, & issue nice commands to them

Actions course: coming soon

Copyright © 2018. All rights reserved.

Version 0.1

Show me your Gophish !



What does it look like, really ?

The screenshot displays the ABADDON web interface. On the left, a terminal window shows the command `python3 manage.py runserver` and subsequent log output, including a `RequestsDependencyWarning` and a message about starting a development server at `http://127.0.0.1:8000/`. The main interface has a red header with the **ABADDON** logo and a user profile icon labeled **pool**. A dark sidebar on the left contains a **MAIN NAVIGATION** menu with items: Dashboard, Reconnaissance, Weaponization, Delivery, Exploitation, and Post-Exploitation. The main content area features an **EC2 deployment dashboard** with a **Submit** button and the text **Deployed Gophish on: i-0f98d2b265**. The footer includes **Copyright © 2018. All rights reserved.** and **Version 0.1**. A system tray at the bottom right shows **Computer Charging (69%)**.

OPSEC thoughts

What Red Teamers should be aware of, what we want to automate

- / The phishing **domain name MUST NOT point to an IP address belonging to the red teamers' organization at any time** during TLS configuration (or it will be logged in that IP's reverse DNS history)
- / **Personal accounts MAY be used** for setting up the AWS infrastructure (domain registration, EC2 instances, Cloudfront, SES...) → your organization must be prepared to answer AWS requests
- / **Do not forget to activate Cloudfront logs** (that's why we create a dedicated s3 bucket)
- / **Use several:**
 - ❑ Domain names for mail addresses & landing page
 - ❑ Payload storage locations

Security products avoidance is hard



Goal: Automate what would happen if the red teamers had penetrated the network



What we've done: a RAT dashboard

What we've automated

- / Receive the RAT connection
- / Launch discovery commands automatically
- / Launch arbitrary commands from the GUI

What could be improved

- / Efficient tools to spot passwords on public shares (not just: `findstr /s /i /p "pwd passw mdp confid securestring" *` or PowerView Find-InterestingFiles)
- / The dashboard GUI 😊
 - Multiple payloads handling
 - Lots of responsive things

Show me your dashboard !



What does it look like, really ?

<input type="checkbox"/>	ID	INFO	TIMING	COMMANDE	RESULTAT
<input type="checkbox"/>	2678276115	W5CG72853TG0817 charles.ibrahim C:\WINDOWS	3		%wmic process list full ----- CommandLine= CSName=W5CG72853TG0817 Description=System Idle Process ExecutablePath= ExecutionState= Handle=0 HandleCount=0

ABADDON

pool

MAIN NAVIGATION

- Dashboard
- Reconnaissance
- Weaponization
- Delivery
- Exploitation
- Post-Exploitation

Title - RAT handler panel

Id: test

Info: test

Command: whoami /all

Output:

Polling Time: 10

Last Request: 2018-11-23 13:06:31

Requestnature: AUTH

Submit

The old C&C GUI

The new one



/ **04** What we wished, what we need, what we'll get

Conclusion

Good workers need fine tools

- / RT operations are technically complex, politically sensitive
- / **Automation is key:** it saves time, avoids silly mistakes, helps doing simply complex tasks
- / The tool may be used to realize separate tasks easily: passive reconnaissance, external cartography, phishing campaigns, etc.

Fine tools need workers

- / A **pretty amount of work** involved, with some actions not really easy to perform (AWS deployment management was not that simple)
- / **So far:** 1 lead dev (~20), 1 other auditor (~15), 2 interns (~10) involved ~ 45 pure dev days of work