Siesta Time v0.2

Implant, Infrastructure and Reporting

Who m I ?

- Spoke at R.T. Vill. Defcon27,Happy to beat Shellcon19!!
- Originally from Spain, Sevilla: “**otra servesa por favoh**!”
- Most of my life/studies in Europe
- OSCP Certified, Aiming for CREST in the future :)
- Product Security in “big CRM” company for the last 3 years
- Si !
  - The “before last” drink, “Salir de tranquis”, memes/gifs/paint!
  - Into the box/Complex problems/**Following white rabbits**
  - Burning man/Rocio, “Feria de Sevilla”
- Meh !
  - Not going out on Saturday night for 6AM hikes on next day
  - “Scratch” security (XSS and friends), “auditing”,checklists…
  - Coachella & Sparkling Ponies, Holy Week

Why this? Disclaimers/advisory

- Personal career interest in red teaming/Offsec (educational)
- Implants+Infrastructure are key on red team ops
- Decided to build my own implant tool (control over ops)
- Strong web/high level programming knowledge(centralized frm.)
- This tool is not new… throwback,sliver,Cobalt Strike (not open source),empire (stopping support??)… But it is new in the idea of gluing all together to create a good Red Team tool/framework!
- Why “Siesta” “Time”? Sleeping is good my hax0rs (while automation happen)
- The tool is still in a very alpha stage, but will be growing! Tons of bugs and problems… Needs help to code!
- Not yet tested in “real” environments…but will be

Let’s figure out the start - Preparing ops. / Infra

We got an engagement **with limited time**

Start to register domains, start to configure the Virtual Private Clouds, Maintain accounts, share it between Operators...

Prepare possible staging servers:

- Empire, MSFT, Cobalt Strike...to receive fast shells

Prepare for possible detection! →

- Redirectors for Long-Hauls Implants
- Big quantity of them, it is difficult to keep them organized

Let’s figure out the start-Developing for ops./Implant

The implant development cycle...

- It needs to egress, how will it egress? (TCP/SSH/HTTP)
- It needs to persist (each method will be “compliant” with target EDR/HIDS),
- What if the egress doesn´t work for target ? Need to re-write, more time
- Do I have the code from the previous one? Can we recycle old code?
- Which are the abilities, how it will interact with OS resources? (Features like native libraries API’s (syscalls/C++ wrappers) vs “os.exec(cmd/bash)”
- Oh rayos! For how many platforms does it need to be compiled?

Let’s figure out the start - It is working!/ AfterMatch!

Sut! We have been detected, more redirectors/implants (extra time)

Got my objectives, how I exfiltrate? More servers/techniques

Now the Client/Blue team is asking us to make more noise!

More time to craft another implant with different egress method abilities

Reporting! → Have we saved all our actions? All commands from Interactive shells? What about the implants configurations themselves? Do we know what exactly triggered HIDS/EDR alarms?

Time, time, time… we could have spent more time on LM for reaching more difficult targets… or at least for a Siesta...

Can we do it better? Can we automate/recycle it?

Is it possible to create a General working Red Team Infrastructure?

- Need a working strong design for ops

Can we automate it to save time? Can we have a nice GUI? Consoles are too hax0rs for me (millennial style)

- Servers organization and deployment automation. Clicks save time.

Could we have a battery of “modules” ready to test different implants and make the blue team “HIDS” to be correctly evaluated (purpling) ?

- Implant Modularity

Objectives

- Help Companies' Red teams to spend less time on building a clean and efficient C2 infrastructure and generate working Implants
- Modularity on implants for testing/recycling purposes
- Deploy staging servers that will provide operators gap to continue the Assessment while saving commands into reporting documents
- Generate reports with all jobs, errors and interactive console per user basics
- This will **help defenders to be ready for real actors.**

Tool Infrastructure - What it should be, and what it is

Hive → The Operation Server, the center of all infrastructure

Operators → Connect to Hive per user basic, interact indirectly with implants and stagings

Redirectors → Redirect jobs directly from implants or SaaS's

Bichito → The main implant, with the objective of persist and keep himself hidden

Stagings → Short-lived server, for post-exploitation and interactive sessions

Reports,logs,…

Tool Infrastructure - Hive and Operators

The tool is installed using a script:

- Create DB
- Compile client and Hive
- Use a config.txt with VPS/Domain data to deploy Hive with terraform

Client.go → LocalHost golang server running on operators devices. It will automatically log into Hive using saved credentials (compiled into binary). Will refresh GUI JSON data and send Jobs to Hive. In the same time will feed Electron Gui with front-end data, and receive Jobs from it. (It will act like some kind of client side redirector)

Electron GUI → Operators will interact with Siesta Time through this. The GUI will be requesting/sending JSON data to client.go server

Client.go <-> Hive.go → “ Cert Pinning HTTPS + Basic Authentication” to Request/Post JSON to Hive. Session are identified by Client-ID so every action is mapped with an Operator login

Tool Infrastructure - DB (Inside Hive)

Implants → [Redirector1,Redirector2...], Network Module, Persistence Module

VPC → AWS/Azure…, AMI, Region, Keys

Domain → GoDaddy/NameCheap…, Keys

SaaS → Analytics,fronting…, Keys/Credentials

Job → Hive/Bot, Name, Time, Status

Staging → Droplet/MSF…, VPS, Domain, AccessLog/SessionLog

Redirectors, Reports, Logs...

Tool Infrastructure - Implants

Operator Trigger the Job “Create Implant”:

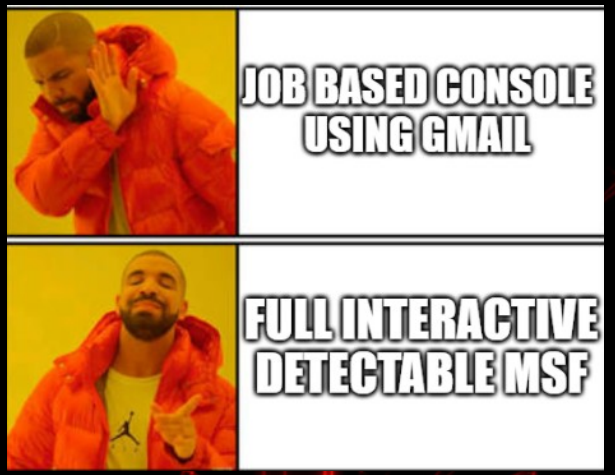
- Select VPS/Domain/SaaS tuples of redirectors
- Select Persistence/Network Module
- Redirector Compilation with target Network Module (Handler)
- Implant Compilation for platforms: OSX, Linux and Windows (x64,x86)
- Deployment with Terraform

Operator sends Jobs to Hive, the ones for Bichitos got redirected to the Redirectors

Bichitos uses Normal Network protocols or SaaS to communicate with redirectors

SaaS: Software as a service, Gmail,Instagram, twitter...

Tool Infrastructure - Stagings



Operator Trigger the Job “Create Staging”:

- Select type of Staging: Droplet, MSF, Empire…
- Select Domain and VPS
- Hive use terraform to deploy and install scripts/services
- Hive install staging service that reverse SSH tunnel Staging Servers to Hive Server

SSH Tunnels let Operator client.go to connect to them through Hive

Droplets are used to drop target executable implants . Server hits are reported/Canaries! (TO-DO)

MSF for example are used to start interactive sessions. All is logged for Reporting!

Implant (Bichito!!) Behaviour-Flow

Implant will first Load Persistence Module (TO-DO)

Next will try to connect to Redirectors:

- If a redirector is not reachable will try next and re-order
- Implant network module will be used for this
- Get Jobs and process them
- Send back results

This will be a repeated behaviour between respTime, if it is not able to connect to Redirectors Implant will die on TTL

Demo:Gmail API Egress

Give me one string… and you will be my C2!

Demo:Reporting

Future Work (a lot) – Working on it

- Security Bugs, Performance Bugs, Native Functions (avoid bash/cmd)
- More Modules for Implant, more SaaS (having a big module battery will be dope)
- More VPS, domains (now just AWS/Go daddy/Gmail working)
- More Detailed reporting (XML?)
- More Stagings (Cobalt Strike?)
- Full GUI Client, Whole user/op management…
- Killchain Generator/Attacks - HTA, Macros (Now just droplets)
- Obfuscation → Choosing between obfuscate or not, etc...
- Root/Persistence, do ever a company test herself with rootkits?
- Google Analytics to egress, bypass TLS fingenprints...

Thanks - Proud to be part of Shellcon

- Please don't let me down! Let’s commit together!
- **CFC → Call for Commits!**
- Thanks to all sources, **previous devs/red teamers with similar ideas**
- To all those golang developers that post in stack overflow
- We are here **to help defenders!** Let’s not forget that
- Questions :)
- Repository:

https://github.com/rebujacker/SiestaTime

Image Slide 4

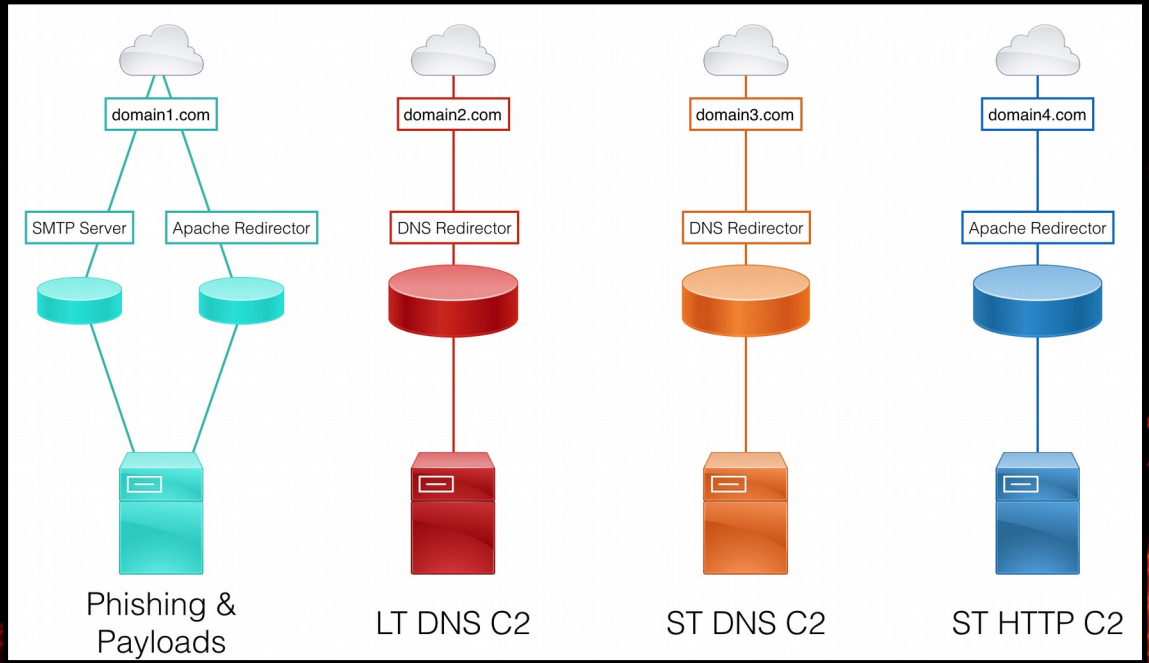


Image Slide 1

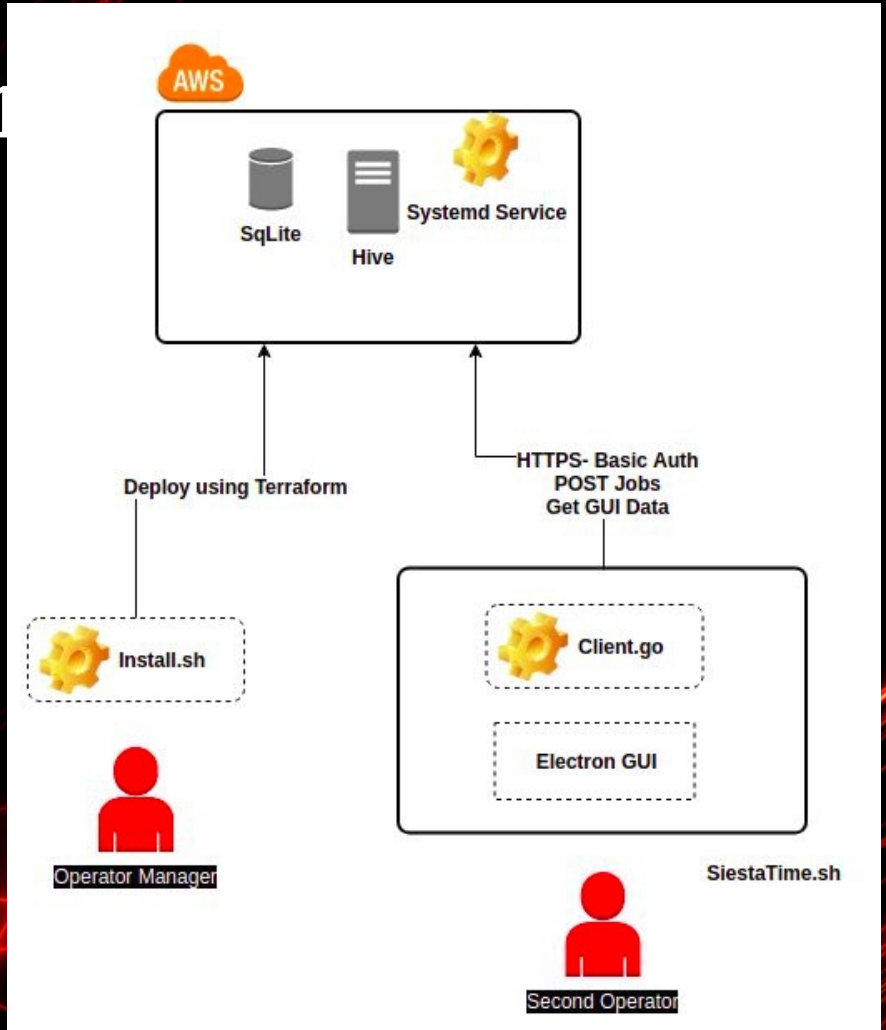
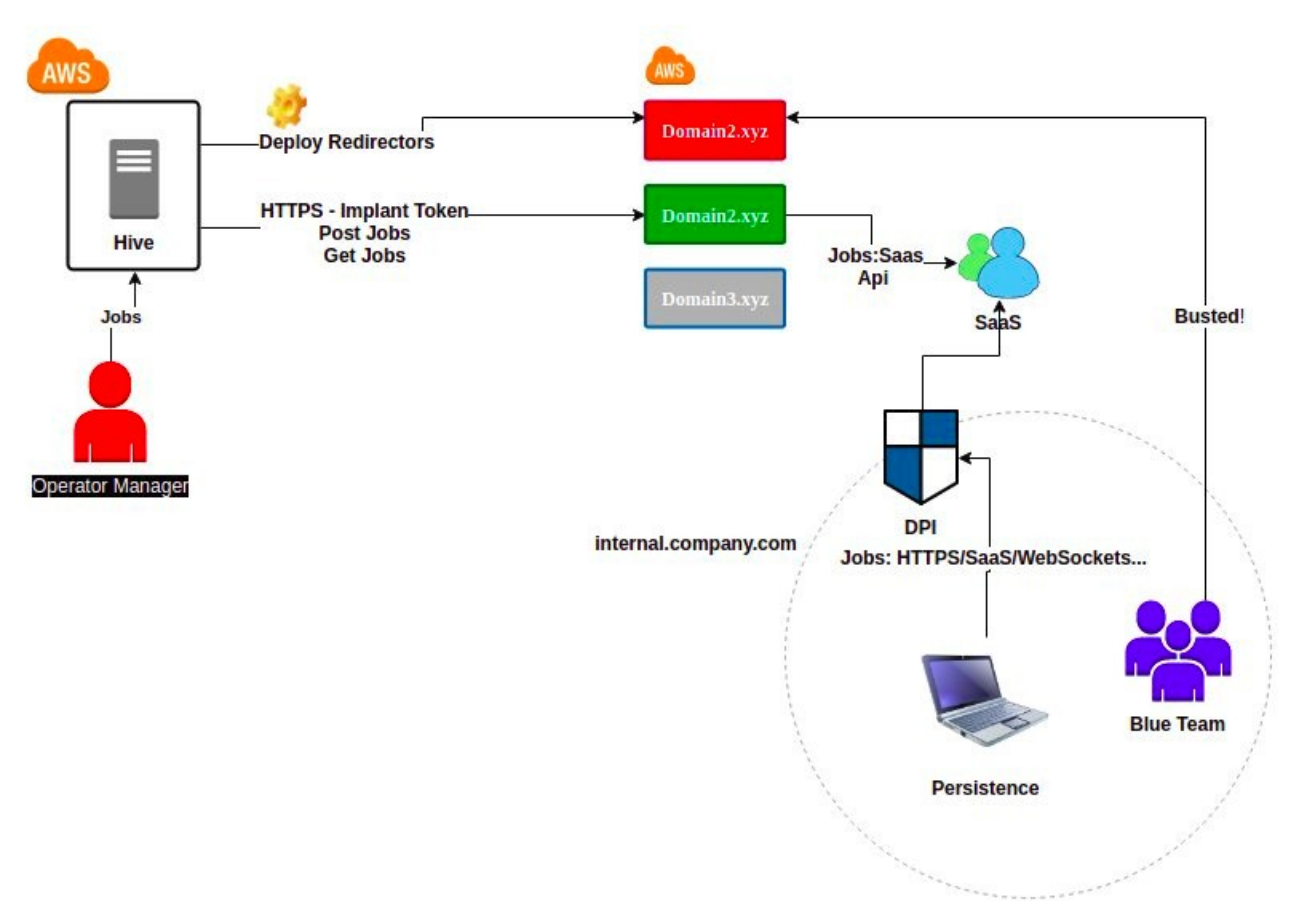


Image Slide 13



Ima

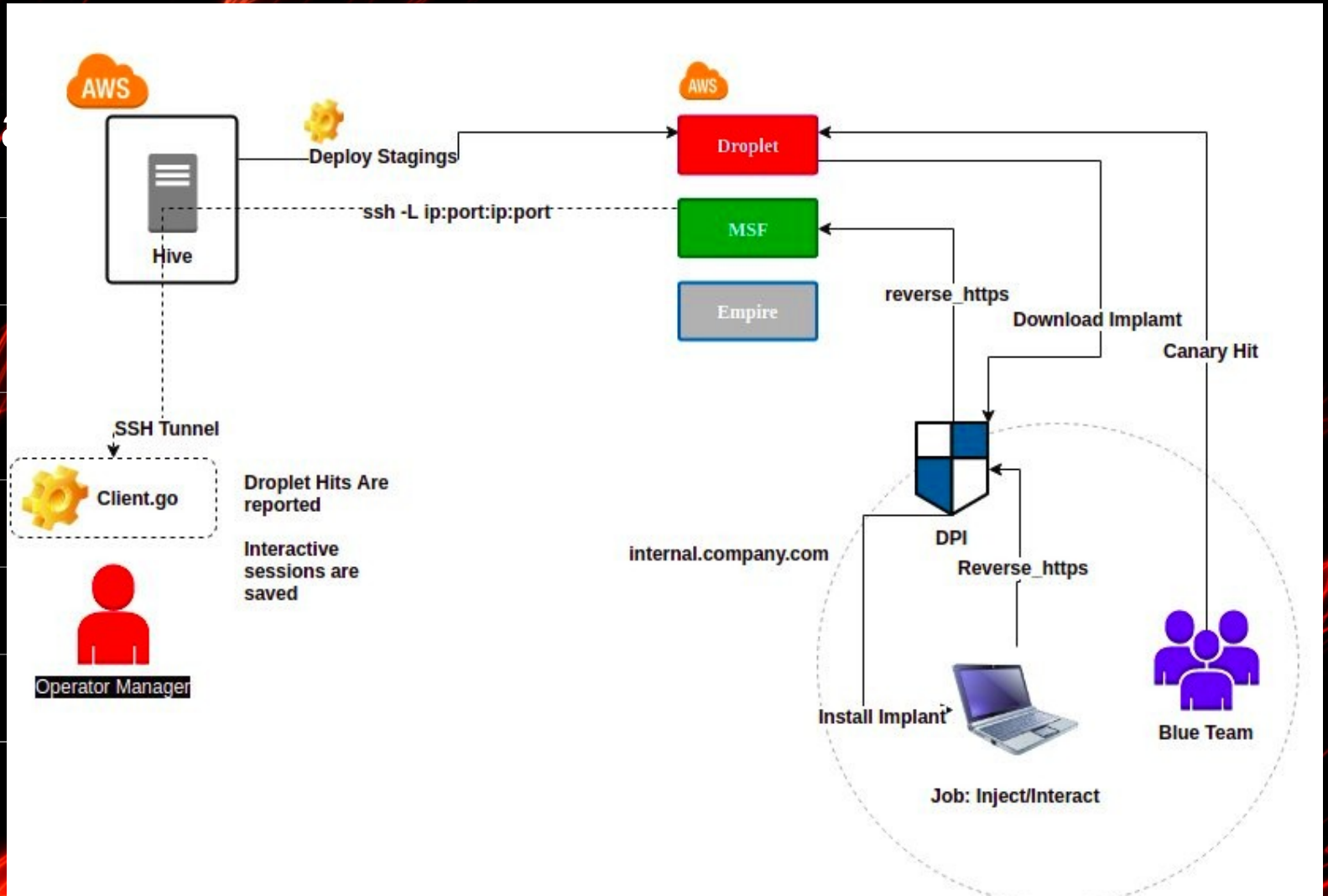

AWS
Hive
Deploy Stagings
AWS
Droplet
ssh -L ip:port:ip:port
MSF
Empire
reverse_https
Download Implamt
Canary Hit
SSH Tunnel
Client.go
Droplet Hits Are reported
Interactive sessions are saved
Operator Manager
internal.company.com
DPI
Reverse_https
Install Implant
Blue Team
Job: Inject/Interact

Image Slide 14