

# Chimaera

Battling Bad Guys  
And Saving Domestic Bliss...

...using Real-Time Web-App Threat Intelligence

# Core Questions

1. How to you secure humans?
  - We won't deal with this today. (WHEW!!)
2. Given the fact that somebody somewhere has an 0day for whatever you're running, how do you stay secure?
  - In other words, how can you fight 0days?
  - In other words, how can you fight advanced/emerging/persistent threats?
  - First person to say "APT" wins the nice pointy hat...

## Answer 1: Detect Compromise

Very promising answer. The thinking goes like this:

1. Most of the effort that goes into security is in aid of detecting attacks.
2. If you accept that attacks will succeed, you can still detect compromises.

There's a lot to talk about here, and Terremark has some very cool projects in this vein, but my talk isn't about them...

## Answer 2: Trickery!

There's already some trickery in play, but...

- Honeypots are great tarpits, but...
  - Generic ones, don't catch the advanced/emerging/persistent stuff
  - Specialized ones, such as custom-built honeyd configurations, are a lot of work
- Reverse Engineering tools are great sandboxes, but...
  - They're designed tricking malware, not humans
  - They're definitely geared for reactive use

Hence...

# Chimaera

Using Honeypot/RE techniques to defend. Two basic steps:

1. Build an illusion for the BG to attack
  - Populate it with realistic targets by replicating real services
  - Transform sensitive data so it looks real but isn't
  - Automate the building of this illusion as much as possible
2. Instrument this illusion
  - Network: packets, flows, etc
  - Hosts: process logs, raw disk/memory access, etc.
  - Code: debugging and introspection to see exploits in action

## OBDifferentiator

- The Chimaera premise is to **mirror actual production services**
- No additional services are simulated, as with honeypots
- The Chimaera approach requires tools and techniques to make this mirroring quick and easy
  - The building blocks for such tools already exist
- The Chimaera strategy is to **divert successful attacks**
- No added vulnerabilities, as with typical honeypot strategies
- No added risk – the attack would have been successful anyway!

## Chimaera's Goals

- Apply lessons, techniques and technologies from RE and Honeypot worlds.
- Specific, targeted applicability
  - I.e., easily support a *particular* application/platform/environment
- General applicability
  - I.e., easily supports *any* particular application/platform/environment
- Relatively quick and easy deployment

A little thinking yielded two promising ideas:

- Toxic VLAN for a network-wise Chimaera
- Reverse-proxied WAF for webapp-wise Chimaera

# CHIMAERA ONE: TOXIC VLAN

# Chimaera One: Toxic VLAN

Typical IR scenario:

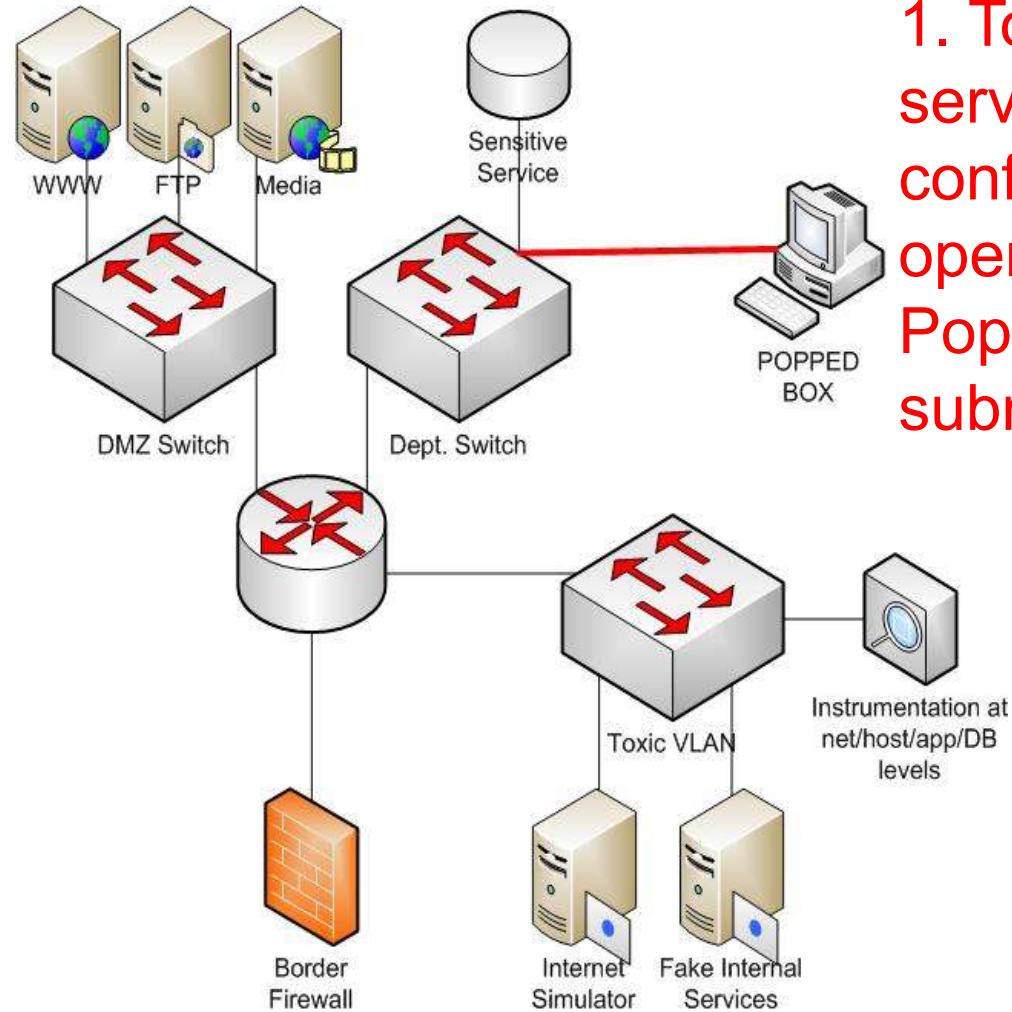
1. BG pops a box and begins lateral infection
2. GG identifies the popped box. Now what?
3. Typically, best case is unplug from the network
  - IOW: declare war
  - BG knows he's blown, begins defensive maneuvers

Result: weekends are lost, domestic bliss is disrupted

# The Toxic VLAN Scenario

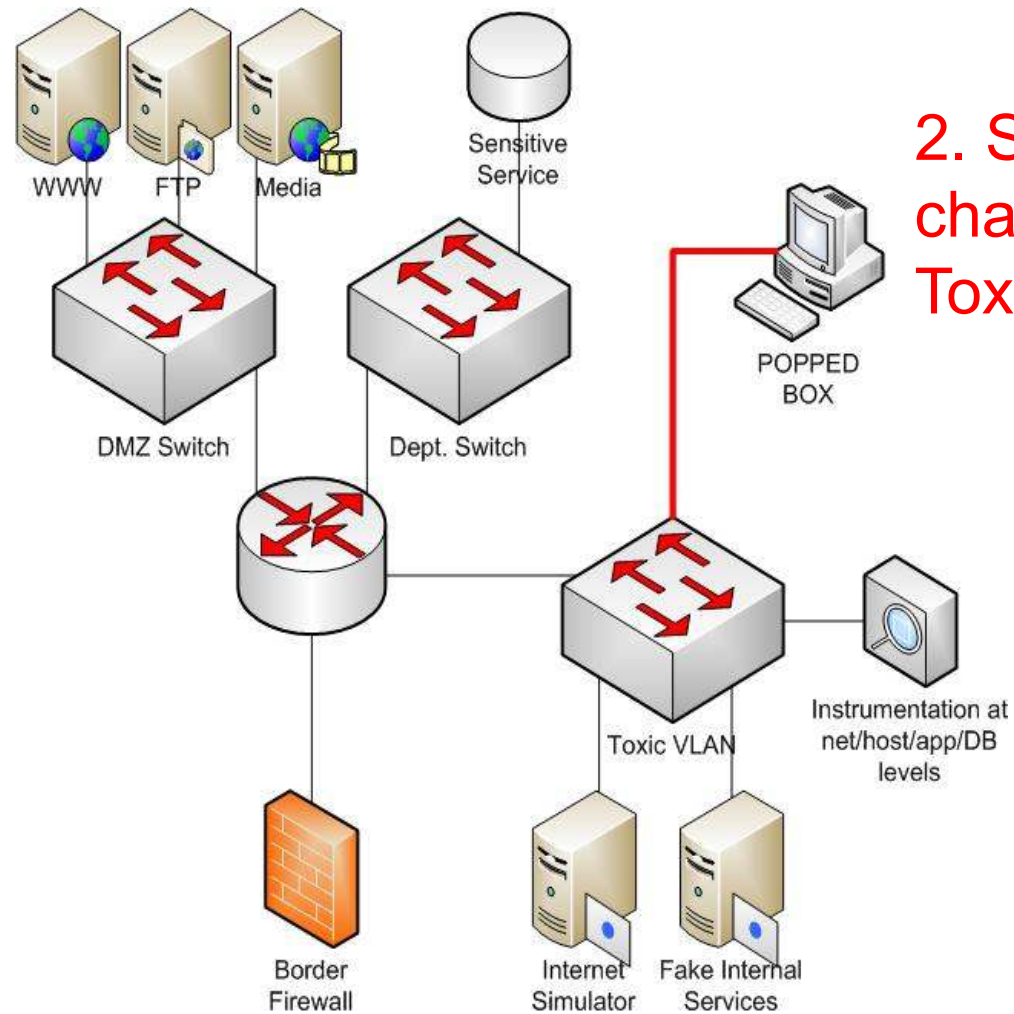
- Basic idea is to build a “copy” of corporate infrastructure
  - Using RE and honey-pot tools to simulate generic services
  - Instrumentation to show low-level details of activity in this VLAN
- Replicate custom/specialized applications
  - Sometimes RE tools and honey-pots won't cut it
  - As more infrastructure is virtualized, replicating it gets easier...
- Transform sensitive data
  - Replace sensitive data with realistic facsimiles
  - Requires that you know where all your sensitive data is...
- Put this fake infrastructure on its own “Toxic” VLAN
- When a machine is popped, put that machine on the Toxic VLAN

# Popped box is identified



1. Toxic VLAN services configured to operate on Popped Box's subnet

# Popped box is contained



2. Switchport changed to Toxic VLAN

# Toxic VLAN and the SANS Six Steps

1. Prepare
2. Identify
3. Contain

## 3.5 Declaw and instrument the Bad Guy

4. Eradicate
5. Recover
6. Lessons Learned

## Domestic bliss is saved

- BG continues “attacking” fake infrastructure
- GG gets intel about malware, tactics, etc.
- GG identifies and contains additional popped boxes
- GG closes initial and secondary vectors (e.g., change passwords, etc.)
- GG restores services and is merely late for dinner
- Meanwhile, BG keeps hacking fake infrastructure!
  - Forever? Well, maybe only for a while
  - Still, good intel for GG, and good tarpit for BG

## Toxic VLAN thoughts

- Reactive defense: contains compromise after a successful attack
- Prevents lateral infection by diverting attacks
- Depends on GG having spare resources
  - To duplicate sensitive services
  - To replace Toxic machines in production
- Danger: Isolate the Toxic VLAN well!
  - And take sensitive data off popped boxes

## Bad news: Toxic VLAN is vaporware

- Ad-hoc versions done after-the-fact during IR
- Hard part is robust replication of “real” environment
  - Lots of services to replicate
  - Lots of sensitive data types to transform
  - Lots of potential popped boxes, so replication has to be configurable to look like the network from any given host’s point of view
- In short, there aren’t tools today to make quick, easy work of this

# CHIMAERA TWO: WAF-RP

## Chimaera Two: WAF-RP

Typical WAF scenario:

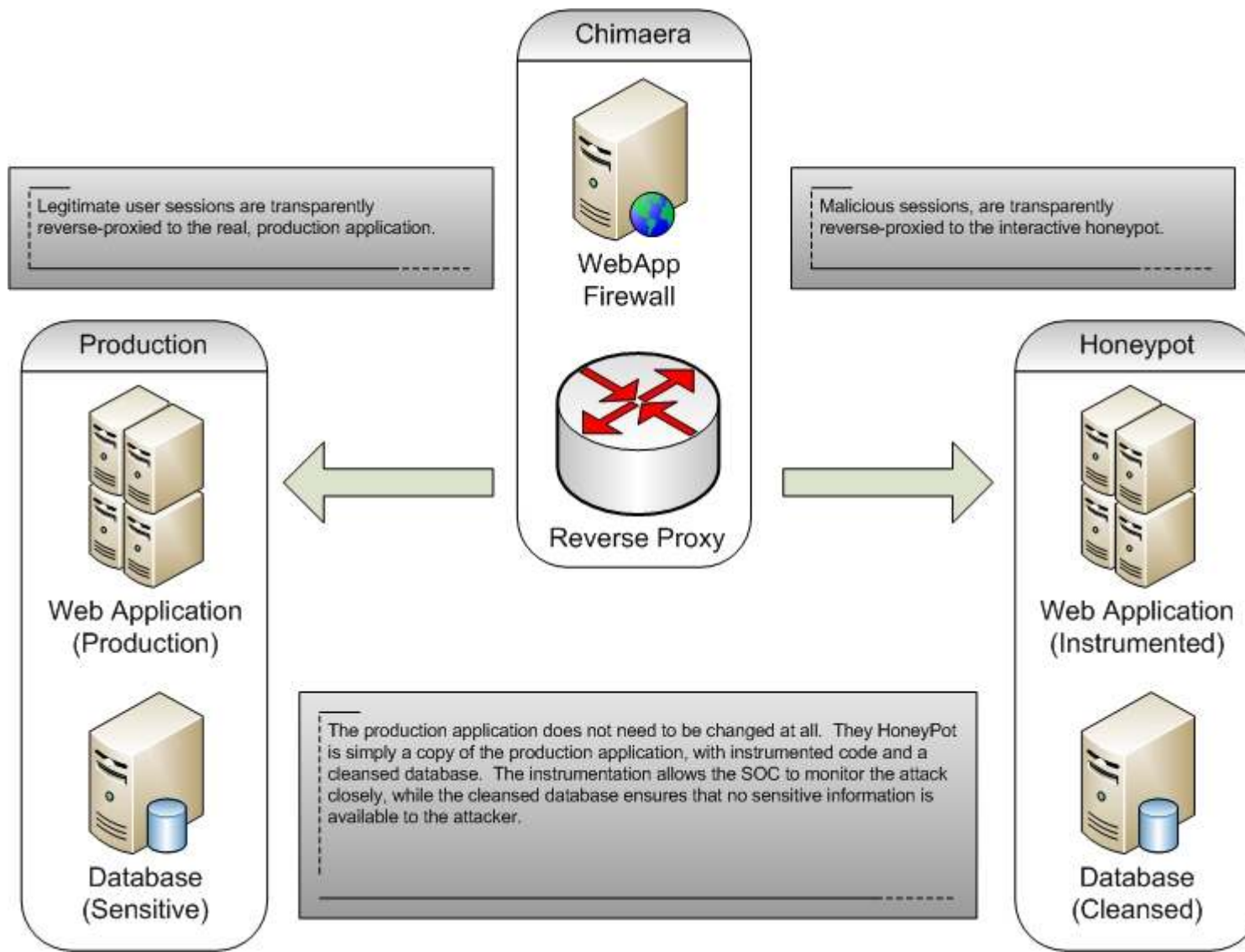
- BG tries lame SQLI tricks, which are blocked by the WAF
- BG keeps trying, using more advanced/emerging/0day tricks until he's in
- GG identifies the popped box at some point

Result: weekends are lost, domestic bliss is disrupted

# The WAF Reverse Proxy Scenario

- Basic idea is to build a “copy” of a web app
  - Using high-interaction honeypot technologies
  - Using automated copy/transform techniques
- Now put a reverse proxy in front of the Real and Fake Applications
- BUT with a way to differentiate between “good” sessions and “bad” sessions
  - Good sessions get proxied to Real App
  - Bad sessions get proxied to Fake App

# Chimaera architecture



## Domestic bliss is saved

- BG continues “attacking” fake web app
- GG gets intel about malware, tactics, etc.
  - We can only hope BG blows his 0days here
- GG tightens up the “real” web app and is merely late for dinner
- Meanwhile, BG keeps hacking fake web app!
  - Forever? Maybe – the BG won’t know what to expect behind the fake web app
  - Consider a Toxic VLAN behind the fake web app...

## WAF-RP thoughts

- Proactive defense: detects attack before a successful compromise
- Diverts – does NOT prevent – compromise
- Depends on BG using lame tricks first...
  - Then you tag that BG's session
  - We'll talk about how to define "session"
- ...OR, depends on BG tripping known IOCs
  - This is the "real-time threat intelligence" piece

## Good news: WAF-RP is demo-ware

- Supports any Apache/PHP web app
  - Uses mod\_security, mod\_header, mod\_proxy
- mod\_header sets session cookie
  - Just one way to track chimaera sessions
  - As distinct from App sessions – next slide
- mod\_security detects lame tricks and IOCs
  - Uses mod\_proxy to transparently reverse proxy to real or fake app based on this differentiation
- HIHAT to duplicate and instrument the web app

# Building the fake web app

- Code instrumentation: HIHAT
  - <http://hihat.sf.net/>: *“allows to transform arbitrary PHP applications into web-based high-interaction Honeypots”*
  - Investigating other options for other languages
- Fake DB with transformed data
  - Identify sensitive fields in DB (SSN, CCN, etc.)
  - Transform these values to realistic fakes
  - Manually scripted today, looking into Talend

# WAF-RP Sessions

- Need a way to identify sessions
  - Many options: IP, agent, headers, resolution, ...
  - <https://panopticklick.eff.org/browser-uniqueness.pdf>
- This is the *Chimaera Session* as distinct from any app session (PHPSESSIONID, etc.)
- Using a cookie for Chimaera state has the benefit of detecting shenanigans
  - Any Chimaera state tampering is bad
  - Any app state is bad, in the absence of Chimaera state

# Making WAF-RP invisible

- Timing side-channel
  - The WAF-RP shouldn't make any difference
  - The instrumented application might run slower
  - Short, random-length pauses at the RP level make this channel very noisy...
- Database equivalence
  - How to mirror updates from real app to fake app
  - In the presence of malicious updates to fake app

## WAF-RP roadmap: real-time DB transforms

- Data in the fake application starts out as a copy of the real DB
  - BUT, with sensitive data transformed to a realistic facsimile
- Real-time updates to fake application are easy
  - DB triggers to update fake DB with transformed data
- BUT what about updates to the fake DB?
  - Updates from the real DB's trigger could overwrite those
  - Or, the triggers could be made smart enough to deal with this
- Trickiest part: automatic trigger generation
  - I **think** smart transform-triggers will be DB- and app-specific
  - The goal is to make Chimaera as easy to deploy as possible, so...

# WAF-RP roadmap: real-time threat intelligence\*

This is really two things:

1. A mechanism for adding simple IOCs to the WAF rule set
  - Source IPs, User Agents, etc., from a centrally-managed repository
  - LUA support in mod\_security could make this trivial
  - Work begun on the centrally-managed IOC repository
2. A way to easily create WAF rules from captured bad traffic
  - A little more complex to implement
  - Frankly, this could be moot if we use...

\* Yes, the name of this talk placed an inordinate amount of emphasis on this particular theme. I blame Microsoft.

## WAF-RP roadmap: white-listed rule set

- Application-specific rule set describing allowed activity
  - E.g., only allow good values for POST variables (trivial)
  - E.g., only allow good page sequences (a little tricky)
- Tools exist to help get a good starting rule set
  - ModProfiler from the mod\_security guys

## WAF weakness



- False positives
  - Even more of a weakness than with regular WAFs
  - Being invisible, you might end up with some confused users...
  - A white-listed rule set can help
- False negatives
  - Less of a weakness than with regular WAFs
  - In a sense, the inevitability of false negatives is the whole point of this exercise...

# Chimaera: End Game

- I'm very interested in community feedback and collaboration
- Discussion about the underlying concepts
  - Sleight-of-hand as defensive philosophy
  - Identifying BGs by their lame first moves
  - Making realistic illusions against which BGs can spend energy
- Discussion about particular implementations
  - Architecture and technology choices
  - Real-time data transformation

**APPLAUSE**

## Questions and Answers

