Simulating Your Way to Security

One Detector at a Time

Slava Nikitin

Workflow

1. Threat model
2. Collect data
3. Explore data
4. Test for discriminability
5. Build a model
6. Fit the model to data
7. Test the estimation algorithm
8. Test the estimate
9. Test the detector
10. Repeat 3-8 or deploy

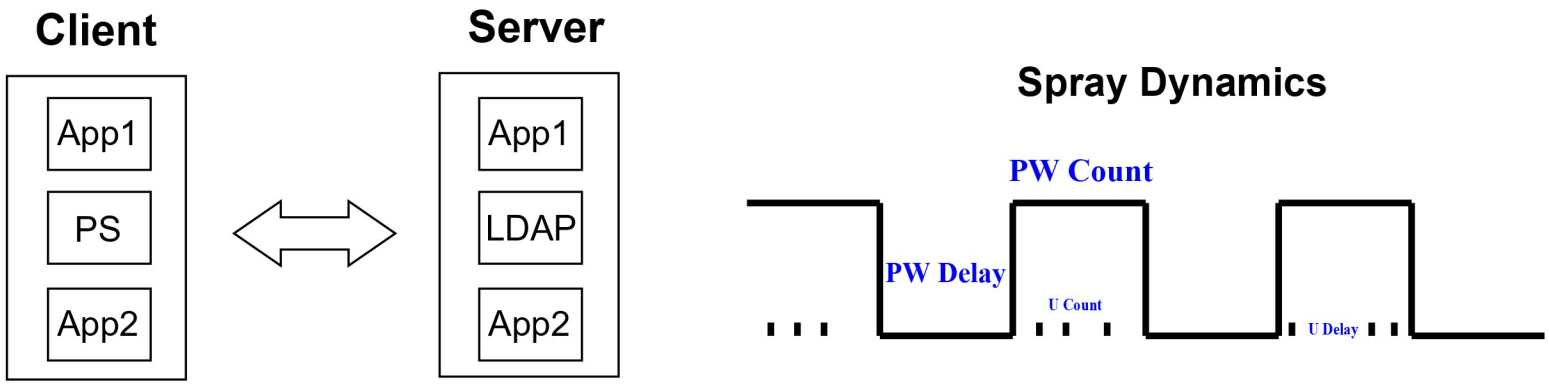
MITRE

Assumption: Attack is relevant

Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Exfiltration	Command And Control
10 items	31 items	56 items	28 items	59 items	20 items	19 items	17 items	13 items	9 items	21 items
Drive-by Compromise	AppleScript	.bash_profile and .bashrc	Access Token Manipulation	Access Token Manipulation	Account Manipulation	Account Discovery	AppleScript	Audio Capture	Automated Exfiltration	Commonly Used Port
Exploit Public-Facing Application	CMSTP	Accessibility Features	Accessibility Features	Binary Padding	Bash History	Application Window Discovery	Application Deployment Software	Automated Collection	Data Compressed	Communication Through Removable Media
Hardware Additions	Command-Line Interface	AppCert DLLs	AppCert DLLs	BITS Jobs	Brute Force	Browser Bookmark Discovery	Distributed Component Object Model	Clipboard Data	Data Encrypted	Connection Proxy
Replication Through Removable Media	Control Panel Items	AppInit DLLs	AppInit DLLs	Bypass User Account Control	Credential Dumping	File and Directory Discovery	Exploitation of Remote Services	Data from Information Repositories	Data Transfer Size Limits	Custom Command and Control Protocol
Spearphishing Attachment	Dynamic Data Exchange	Application Shimming	Application Shimming	Clear Command History	Credentials in Files	Network Service Scanning	Logon Scripts	Data from Local System	Exfiltration Over Alternative Protocol	Custom Cryptographic Protocol
Spearphishing Link	Execution through API	Authentication Package	Bypass User Account Control	CMSTP	Credentials in Registry	Network Share Discovery	Pass the Hash	Data from Network Shared Drive	Exfiltration Over Command and Control Channel	Data Encoding
Spearphishing via Service	Execution through Module Load	BITS Jobs	DLL Search Order Hijacking	Code Signing	Exploitation for Credential Access	Password Policy Discovery	Pass the Ticket	Data from Removable Media	Exfiltration Over Other Network Medium	Data Obfuscation
Supply Chain Compromise	Exploitation for Client Execution	Bootkit	Dylib Hijacking	Component Firmware	Forced Authentication	Peripheral Device Discovery	Remote Desktop Protocol	Data Staged	Exfiltration Over Physical Medium	Domain Fronting
Trusted Relationship	Graphical User Interface	Browser Extensions	Exploitation for Privilege Escalation	Component Object Model Hijacking	Hooking	Permission Groups Discovery	Remote File Copy	Email Collection	Scheduled Transfer	Fallback Channels
Valid Accounts	InstallUtil	Change Default File Association	Extra Window Memory Injection	Control Panel Items	Input Capture	Process Discovery	Remote Services	Input Capture		Multi-hop Proxy
	Launchctl	Component Firmware	File System Permissions Weakness	DCShadow	Input Prompt	Query Registry	Replication Through Removable Media	Man in the Browser		Multi-Stage Channels
	Local Job Scheduling	Component Object Model Hijacking	Hooking	Deobfuscate/Decode Files or Information	Kerberoasting	Remote System Discovery	Shared Webroot	Screen Capture		Multiband Communication
	LSASS Driver	Create Account	Image File Execution Options Injection	Disabling Security Tools	Keychain	Security Software Discovery	SSH Hijacking	Video Capture		Multilayer Encryption
	Mshta	DLL Search Order Hijacking	Launch Daemon	DLL Search Order Hijacking	LLMNR/NBT-NS Poisoning	System Information Discovery	Taint Shared Content			Port Knocking
	PowerShell	Dylib Hijacking	New Service	DLL Side-Loading	Network Sniffing	System Network Configuration Discovery	Third-party Software			Remote Access Tools
	Regsvcs/Regasm	External Remote Services	Path Interception	Exploitation for Defense Evasion	Password Filter DLL	System Network Connections Discovery	Windows Admin Shares			Remote File Copy
	Regsvr32	File System Permissions Weakness	Plist Modification	Extra Window Memory Injection	Private Keys	System Owner/User Discovery	Windows Remote Management			Standard Application Layer Protocol
	Rundll32	Hidden Files and Directories	Port Monitors	File Deletion	Replication Through Removable Media	System Service Discovery				Standard Cryptographic Protocol
	Scheduled Task	Hooking	Process Injection	File System Logical Offsets	Securityd Memory					Standard Non-Application Layer Protocol
	Scripting	Hypervisor	Scheduled Task	Gatekeeper Bypass	Two-Factor Authentication Interception					Uncommonly Used Port
	Service Execution	Image File Execution Options Injection	Service Registry Permissions Weakness	Hidden Files and Directories						Web Service
	Signed Binary Proxy Execution	Kernel Modules and Extensions	Setuid and Setgid	Hidden Users						
	Signed Script Proxy Execution	Launch Agent		Hidden Window						
	Source			HISTCONTROL						
	Space after Filename			Image File Execution Options Injection						

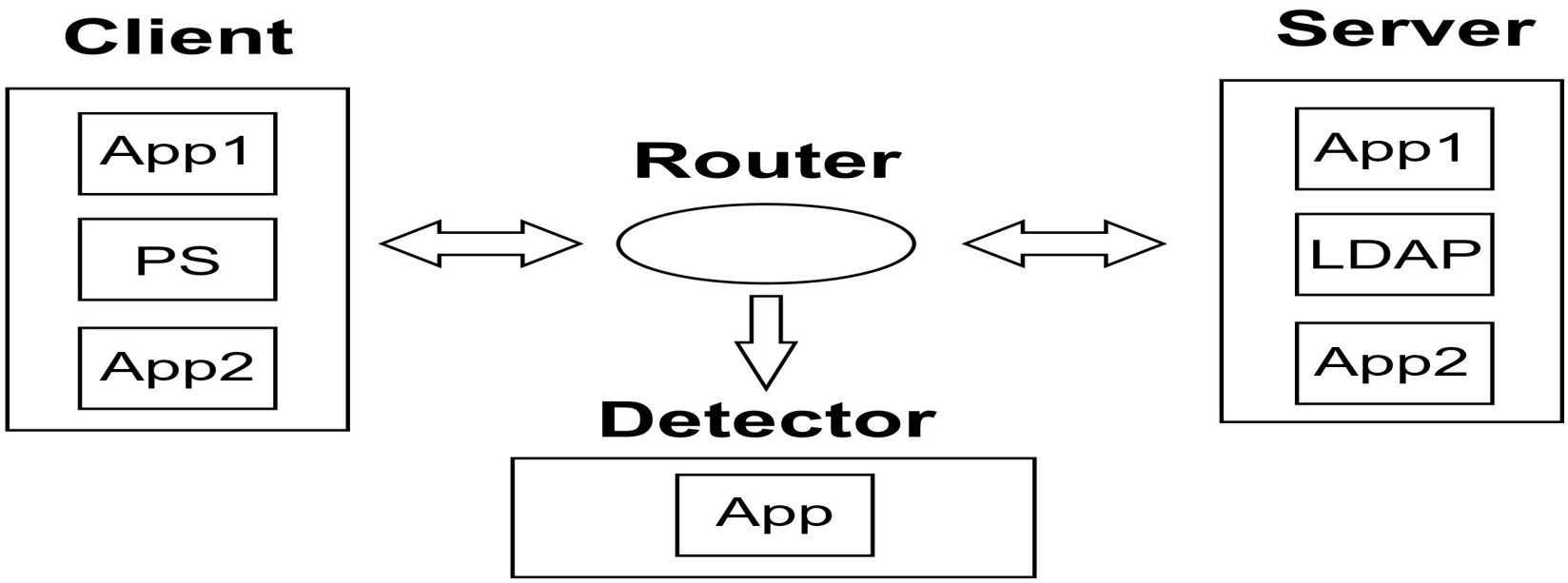
Password Spray Diagrams

Assumption: Attack has a repeatable structure



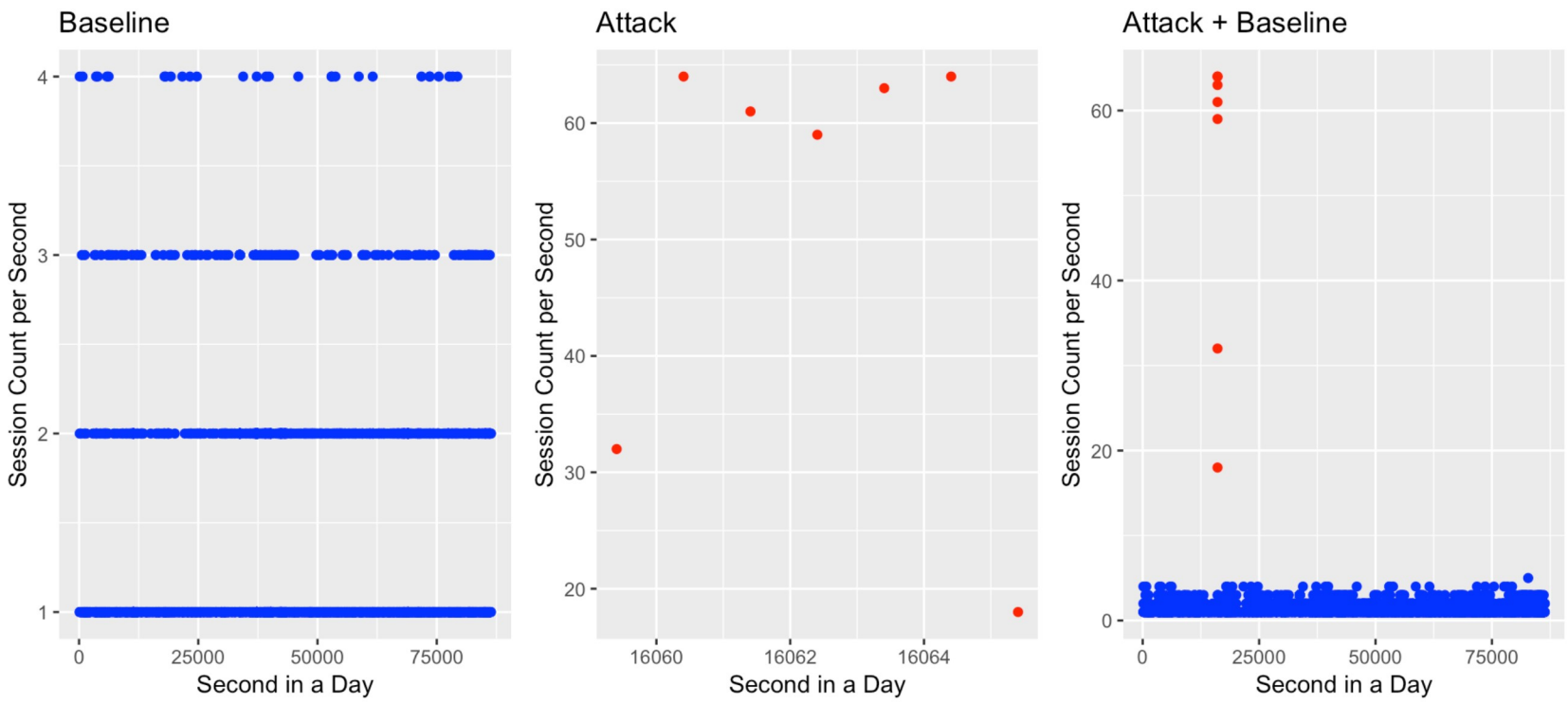
Data Collection Diagram

Assumption: Data sensitive to the attack is captured



Data Exploration

Assumption: Discriminability of attack + baseline from baseline



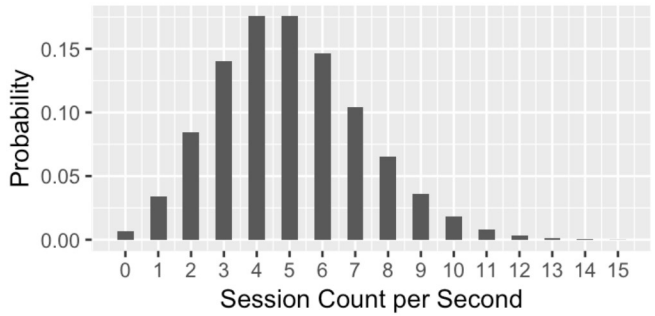
Model

Assumption: Model respects restrictions of the data

$$M = \{p_\theta(y \mid \mathbf{x}) : \theta \in \Theta \subseteq \mathbb{R}^p\}$$

$$Y \in \{0, 1, \dots\}, x_1 \in \{1, 2, \dots, 86400\}, x_2 \in \{1, 2, \dots, 7\}$$

$$p_\theta(y \mid x) = \frac{\exp\{-f(\mathbf{x}, \theta)\} f(\mathbf{x}, \theta)^y}{y!}$$



Estimation Algorithm

Assumption: Algorithm can consistently and accurately recover the best estimate

Penalized maximum likelihood

$$\hat{\theta} = \mathrm{argmax}_{\theta \in \Theta} \left(\sum_{i=1}^{n} \log p(y_i \mid f(\mathbf{x_i}, \theta)) + \lambda g(\theta) \right)$$

parameter	estimate	standard error
(Intercept)	0.3109458	0.0011767
wdayTue	0.0124353	0.0038364
wdayWed	-0.0217065	0.0026326
wdayThu	-0.0504998	0.0030671
wdayFri	-0.0424501	0.0031398
wdaySat	0.0963773	0.0026377

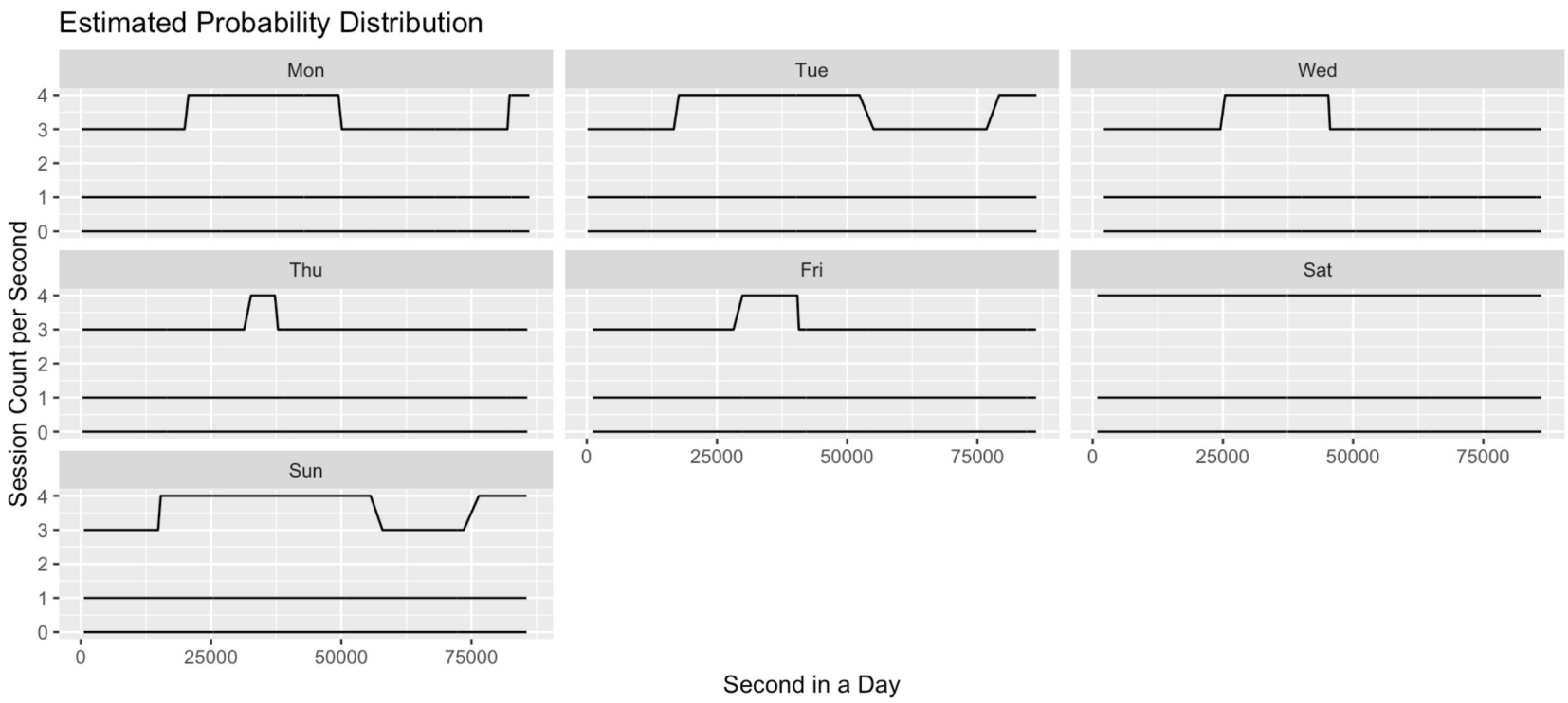
Estimation Algorithm Test

```
diffs <- vector("double", k)

for (i in 1:k) {
  theta_i <- rmvn(theta_hat, sigma_hat)
  estimate$theta_hat <- theta_i

  x_i <- simulate(estimate)
  phi_hat <- estimator(x_i)

  diffs[i] <- phi_hat - theta_i
}

mean(diffs) / sd(diffs)
```

parameter	mean / standard error ratio
(Intercept)	0.315
wdayTue	-0.165

Estimate

Assumption: Estimate can replicate observed data



Estimate test

```
sim_means <- vector("double", k)

for (i in 1:k) {
  x_i <- simulate(estimate, n)
  sim_means[i] <- mean(x_i)
}

obs_mean <- mean(y)
diffs <- sim_means - obs_mean

mean(diffs) / sd(diffs)
```

mean	standard deviation	99% quantile
-0.0528694	32.79969	9.411476

Detector

Assumption: Detector has high TP and low FP for the attack of interest

$$\delta_{\mathbf{x}}(\mathbf{y}) = \begin{cases} 1 \text{ if } y \in D_{\mathbf{x}} \\ 0 \text{ else} \end{cases}$$

$$D_{\mathbf{x}} = \{y : \mathrm{P}(Y > y \mid f(\mathbf{x_i}, \hat{\theta})) < \tau\}$$

Detector Test

$$A_x = \{\mathbf{y} : y_x \in D_x\}$$

$$\text{True Positive: P}\left(\bigcup_{\mathbf{x}} A_x | \text{attack and baseline}\right)$$

$$\text{False Positive: P}\left(\bigcup_{\mathbf{x}} A_x | \text{baseline}\right)$$

Experimental result

Experimental Factors:

$$\text{Time} \in \{\text{None}, \text{Early}, \text{Middle}, \text{Late}\}$$
$$\text{Day} \in \{\text{M}, \text{T}, \ldots, \text{S}\}$$

condition	probability of detection
baseline	0
baseline + attack	1

Conclusions

1. Is the threat relevant?
2. Is the data sensitive to the attack?
3. Is the model constrained to data?
4. Is the attack + baseline traffic discriminable from the baseline traffic?
5. Is our estimation algorithm accurate?
6. Is our estimate accurate?
7. Is our detector accurate?