Proactive and Reactive Security

John Mitchell
Stanford

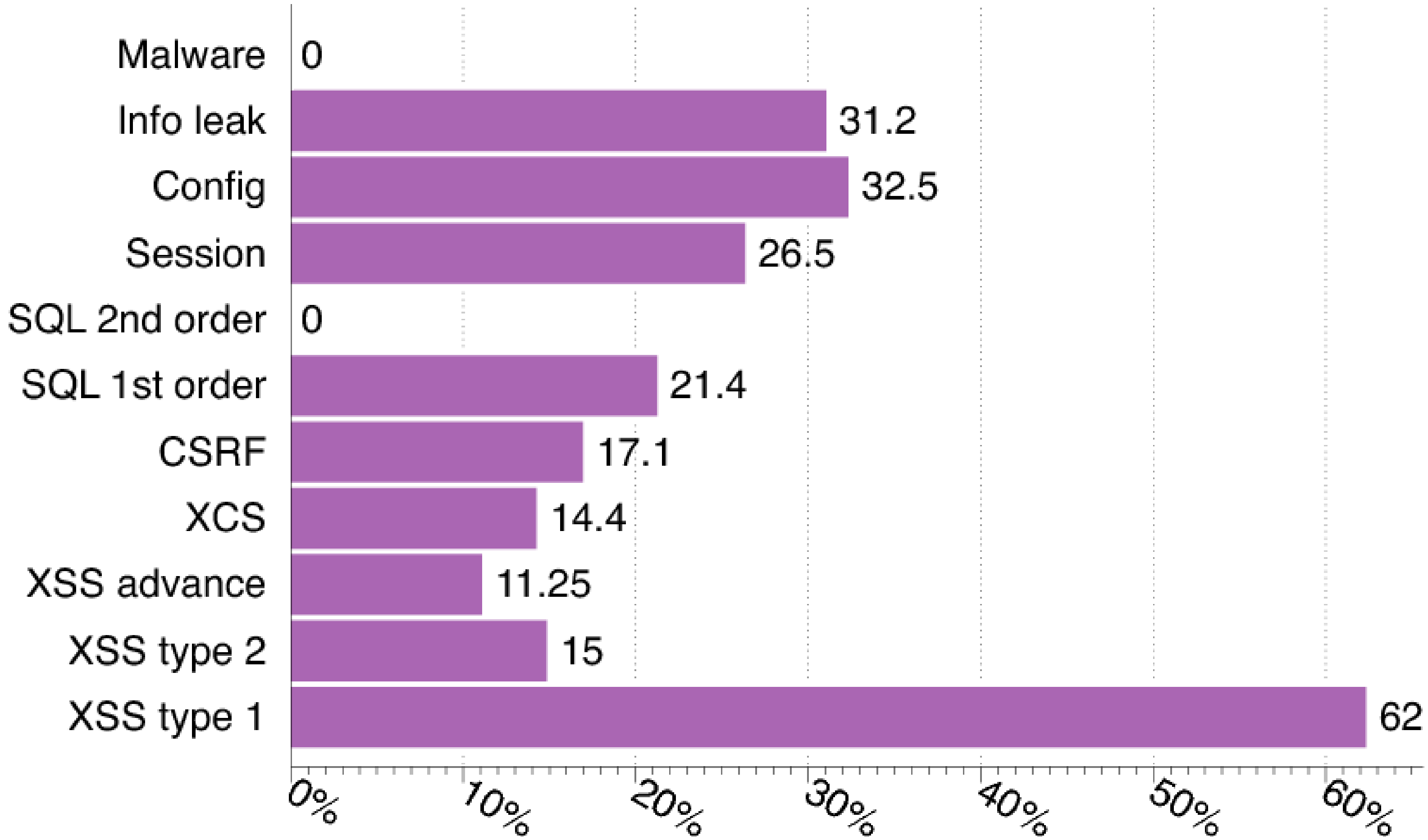
Stanford Computer Security Lab

Study of DNSSEC

Security Property Violations	Prevent At	Prevention Advice
Resource Record remains valid in local resolver cache after expiration of signatures or key rollover (revocation) higher in attestation chain	Resolver Software ISP	Resolver software sets RR TTL to depend on all signatures in attestation chain to trust anchor Resolver software imposes an independent (from authoritative zone values) cap on TTL and signature validity periods
Glue records may be forged to direct next recursive query to attack DNS server	Domain Operator Resolver Software	Use all secure delegations If forgery is suspected, query supposed authoritative zone to obtain signed version of glue records. (Even if no action is taken, this violation does not result in acceptance of forged RR as final query answer. See paper section.)
NSEC3 opt-out may be used to prepend falsified owner name in domain, as stated in RFC 5155, resulting in vulnerability to cookie-theft and pharming	Domain Operator Website Designer	Do not set NSEC3 opt-out flag Do not use overly coarse cookie "domain" setting
Replay of still valid A+RRSIG after IP-address move (Bernstein [12])	Domain Operator	Do not relinquish IP-address until all A+RRSIGs have expired
Inter-operation with standard-DNS child zones means insecure answer returned by DNSSEC resolver	Domain Operator	Adoption of DNSSEC; Do not interoperate DNSSEC with DNS
Lack of end-user software indicator of secure vs. insecure DNSSEC query result exposes end-user to exploitable insecure DNSSEC query result	End-User Software (Browser/OS)	Support DNSSEC by providing lookup security indicators using DNSSEC AD Bit
Network attacker can arbitrarily manipulate DNSSEC reply header and status bits	Resolver Software ISP or OS	Do not trust header bits. Resolver validates only using internal state and signed RRs. Cannot trust remote DNSSEC validation without secure channel. Provide secure channel or validate all DNSSEC RRs locally
Network attacker can add recorded RRs / subtract RRs / mangle bits in RRs in DNSSEC reply packet	Resolver Software	Build *attested cache* for answering user queries using only authoritative signed RRs contained in DNSSEC replies.

Black-box testing tools

Overall detection rate for all tested scanners

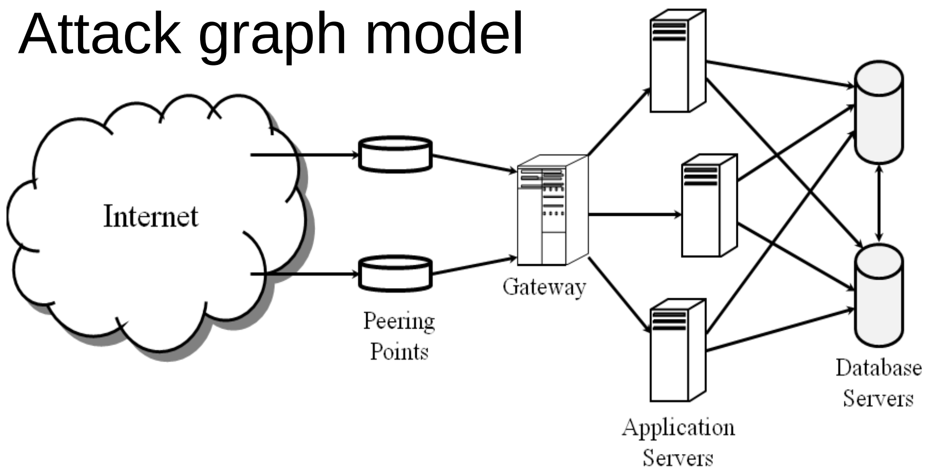


Commercial tools on custom evaluation site

Reactive vs proactive security

- Problem: Can a reactive security strategy be an effective defense?
- Solution: Game-theoretic analysis, using multiplicative update method from economics
  - Intuition: attacker is like an "investment expert" who tells defender where to invest in defenses
  - Discount past attacks exponentially to achieve strategy competitive with proactive benchmark

Attack graph model



- Reactive strategy performs as well as proactive if
  - No catastrophic attacks
  - Defense budget is fungible
- Reactive requires less information
  - Need not know entire attack graph
  - Need not know attacker's utility function

A Learning-Based Approach to Reactive Security

A. Barth, B. Rubinstein, M. Sundararajan,

J.C. Mitchell, D. Song, and P. Bartlett

Financial Cryptography and Data Security '10

Reactive Security

- How effective is reactive security?
- Some factors that matter:
  - Low probability catastrophic attacks
  - Liquid budget vs large one-time expenses
  - Attack costs linear in defense investments
- One positive result
  - Under certain assumptions, reactive defense is competitive with best proactive strategy

How could this be useful?

- Proactive Security > Reactive Security?
  - Proactive ⇒ predict the future ⇒ hard
  - Reactive ⇒ learn from the past ⇒ easier
- Enterprises must allocate limited resources
  - How much is it worth paying to find attacks in advance?
  - For which kind of threats?

Framework

- Game-based model
  - "Attack graph" model, some parts known to attacker but not defender
- Study competitive "return on attack" (ROA)
  - Learning-based reactive strategy to be competitive with the best fixed proactive defense
  - Competitive ratio

$$(\text{proactive ROA})/(\text{reactive ROA}) \leq 1 + \varepsilon,$$

provided the game lasts $\Omega(1/\varepsilon)$ rounds

Proof leverages multiplicative update method, used in investment theory

Adversarial Metrics

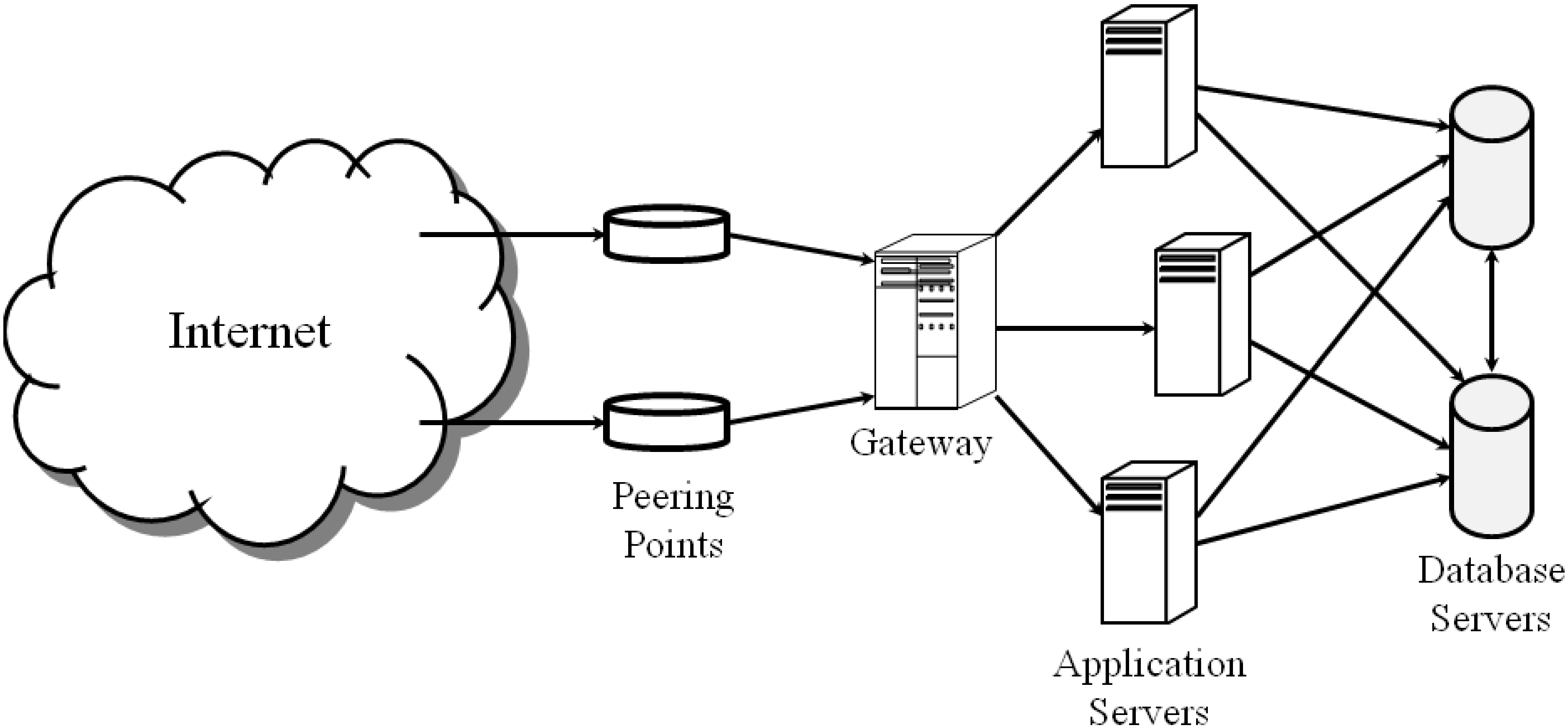
- Attacker profit

  payoff from attack – cost to mount attack

- Return-on-attack

  payoff from attack / cost to mount attack

Attack graph



Model: Directed Graph

- Edges provide an "attack surface"
  - Bigger attack surface ⇒ costs more to defend
  - Defender allocates budget over edges
- Nodes have value (to the attacker)
- Attacker selects attack path
  - Pays to cross each edge, gains rewards along the way

Repeated Game

- In each round t from 1 to T:
  - The defender chooses defense allocation $d_t(e)$ over the edges $e \in E$
  - The attacker chooses an attack path a_t in G
  - The path a_t and attack surfaces $\{w(e) \mid e \in a_t\}$ are revealed to the defender
  - The attacker pays $cost(a_t, d_t)$ and gains $payoff(a_t)$

Learning Reactive Strategy

- Intuition: Reinforce edges used in attacks
  - Shift defense budget to counter observed attacks
  - Harder for attacker to mount same attack
- How quickly to re-allocate budget?
  - Too fast: attacker can cycle through lucrative attacks
  - Too slow: attacker can repeat lucrative attacks

Technique: Experts Learning

- Algorithm from online learning theory
- Multiplicative update
- Property: Regret minimization

Reactive Strategy

Algorithm 1 A reactive defense strategy for hidden edges.

- Initialize $E_0 = \emptyset$
- For each round $t \in \{2, ..., T\}$
  - Let $E_{t-1} = E_{t-2} \cup E(a_{t-1})$
  - For each $e \in E_{t-1}$, let

$$S_{t-1}(e) = \begin{cases} S_{t-2}(e) + M(e, a_{t-1}) & \text{if } e \in E_{t-2} \\ M(e, a_{t-1}) & \text{otherwise.} \end{cases}$$

$$\tilde{P}_t(e) = \beta_{t-1}^{S_{t-1}(e)}$$

$$P_t(e) = \frac{\tilde{P}_t(e)}{\sum_{e' \in E_t} \tilde{P}_t(e')} \ ,$$

where $M(e, a) = -\mathbf{1}\left[e \in a\right]/w(e)$ is a matrix with $|E|$ rows and a column for each attack.

Positive Result

Theorem 1 *The average attacker profit against Algorithm 1 converges to the average attacker profit against the best proactive defense. Formally, if defense allocations* $\{d_t\}_{t=1}^T$ *are output by Algorithm 1 with parameter sequence* $\beta_s = \left(1 + \sqrt{2\log|E_s|/(s+1)}\right)^{-1}$ *on any system* $(V, E, w, \text{reward}, s)$ *revealed online and any attack sequence* $\{a_t\}_{t=1}^T$, *then*

$$\frac{1}{T}\sum_{t=1}^{T}\text{profit}(a_t, d_t) - \frac{1}{T}\sum_{t=1}^{T}\text{profit}(a_t, d^\star) \leq B\sqrt{\frac{\log|E|}{2T}} + \frac{B(\log|E| + \overline{w^{-1}})}{T} ,$$

for all proactive defense strategies $d^\star \in \mathcal{D}_{B,E}$ *where* $\overline{w^{-1}} = |E|^{-1}\sum_{e\in E} w(e)^{-1}$, *the mean of the surface reciprocals.*

Return on Attack Is Similar

- For all sequences of attacks,

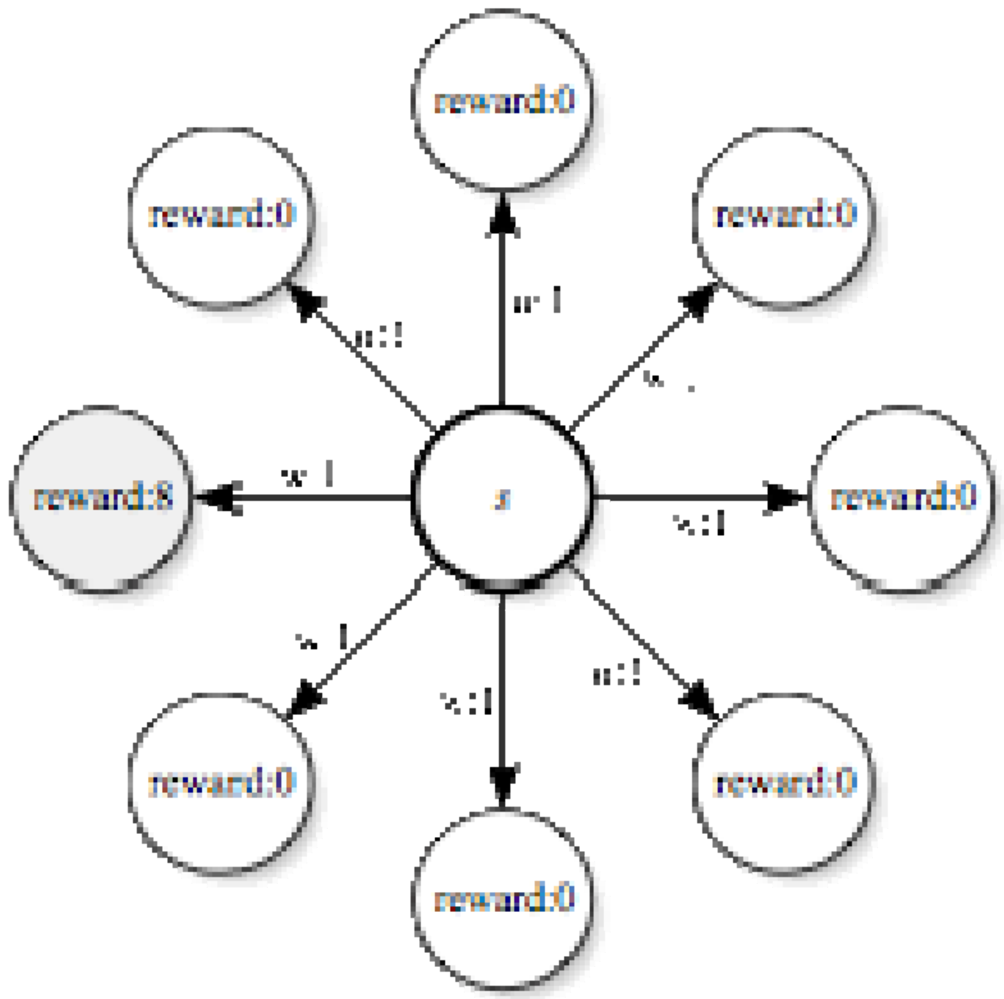
$$\frac{\mathrm{ROA}\left(\{a_t\}_{t=1}^T, \{d_t\}_{t=1}^T\right)}{\mathrm{ROA}\left(\{a_t\}_{t=1}^T, d^\star\right)} \leq 1 + \alpha$$

  – For all α > 0, for sufficiently large T

Generalizations

- Reactive requires less information
  - Need not know entire attack graph
  - Need not know attacker's utility function
- Replace "graph" with "Datalog program"
  - Edges become inference rules
  - Attacks become proofs

Reactive Sometimes Better



Conclusions

- Don’t discount reactive security
  - Observing attacks can be useful
  - Avoid myopic reactive strategies
- Consider investing in agility and monitoring
  - Instead of finding yet more vulnerabilities
  - Defend against real (not theoretical) attacks
- Some threats and defenses outside this model
  - Low probability catastrophic attacks
  - Some defensive investments are not fungible

Result: Reactive ≥ Proactive - ε

- For all sequences of attacks,

$$\frac{1}{T}\sum_{t=1}^{T}\text{profit}(a_t, d_t) - \frac{1}{T}\sum_{t=1}^{T}\text{profit}(a_t, d^\star) \leq B\sqrt{\frac{\log|E|}{2T}} + \frac{B(\log|E| + \overline{w^{-1}})}{T}$$

  – d* is the best possible *time-independent* defense
  – T is the number of rounds, B is defender budget, E is edge count