WRAITS 2009

Lisbon, Portugal

Quantitative Approach to Tuning of a Time-Based Intrusion-Tolerant System Architecture

Presented by: Arun Sood Ph. D.

Co-Author: Quyen Nguyen

George Mason University

Department of Computer Science and International Cyber Center

http://cs.gmu.edu/~asood/scit

703.347.4494

Research supported by contracts from Lockheed Martin and Virginia Center for Innovative Techonogies (partner: Northrop Grumman)

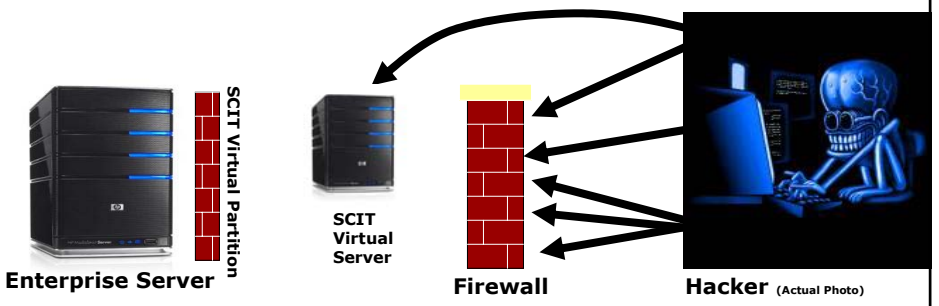
Outline

- SCIT Approach
- MTTSF Derivation
- MTTSF and SCIT Online Window
- Conclusion

The SCIT Solution

Intrusion Tolerance allows malware and hackers into a server…

…but uses virtualization to restore the OS and application to a pristine state after attack!



Every 55 seconds SCIT software cleans and restores the virtual server to its pristine state

Intrusion Tolerance

- **Introducing SCIT, the Intrusion Tolerance System**
  - Optimizes application-specific exposure windows (AEW)
- **Targets "overexposed" applications (transactions)**
  - Servers are sitting ducks
  - Focus initially on Websites, DNS, Single Sign On
  - Ongoing R&D Authentication (LDAP), Firewall
  - Not targeted at applications with inherently long transaction times (FTP, VPN, etc)
- **Leverages virtualization technology to reduce intrusion risk and costs**
  - Reduces exposure time to limit intrusion losses
  - Adds time-based exposure control to intrusion prevention and detection solutions
    - SCIT is based on a new paradigm, but is easy to integrate with existing systems
  - New level of "Day-Zero" protection
- **Increases security through real-time server rotation and cleansing:**
  - Enhances security of high availability systems
  - Enables more flexible patch scheduling

SCIT Software

- SCIT deploys on existing servers - does not require additional physical servers
- SCIT is cost effective, uses virtualization technology and increases system security
- SCIT does not interfere with existing IPS and IDS solutions
- SCIT is an additional layer of defense

How Does SCIT Provide Additional Security?

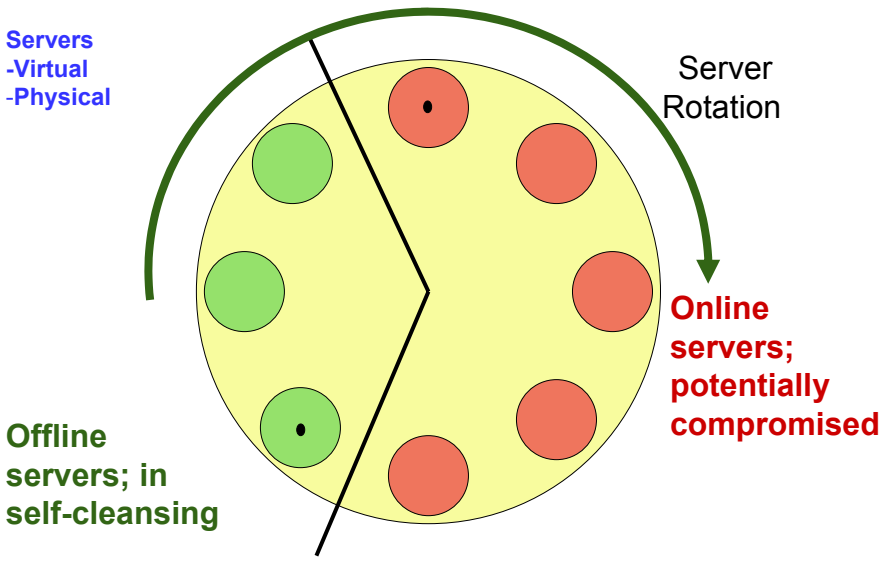
- SCIT servers
  - Regularly restored to a known state and remove malicious software installed by attackers.
  - Provide protection while manufacturer is developing a patch, i.e. SCIT servers are protected in the time period between vulnerability detection and patch distribution.
  - Gives data center managers an additional level of freedom in developing a systematic plan for patch management.
- SCIT DNS servers
  - Domain name / IP address mapping is protected from malicious alteration, thus avoiding improper redirection of the traffic.
- SCIT Web servers
  - Protect the corporate crown jewels, front ends for sensitive information, e.g. customer or employee data sets, IP, and informational web sites.
  - Regularly restores the sites to known states, and makes it difficult for intruders to undertake harmful acts such as deleting files.
  - Avoid long term defacements.
  - Reduces the risk of large scale data ex-filtration.

Comparison of IDS, IPS, IT

Issue	Firewall, IDS, IPS	Intrusion tolerance
Risk management.	Reactive.	Proactive.
A priori information required.	Attack models. Software vulnerabilities. Reaction rules.	Exposure time selection. Length of longest transaction.
Protection approach.	Prevent all intrusions. Impossible to achieve.	Limit losses.
System Administrator workload.	High. Manage reaction rules. Manage false alarms.	Less. No false alarms generated.
Design metric.	Unspecified.	Exposure time: Deterministic.
Packet/Data stream monitoring.	Required.	Not required.
Higher traffic volume requires.	More computations.	Computation volume unchanged.
Applying patches.	Must be applied immediately.	Can be planned.

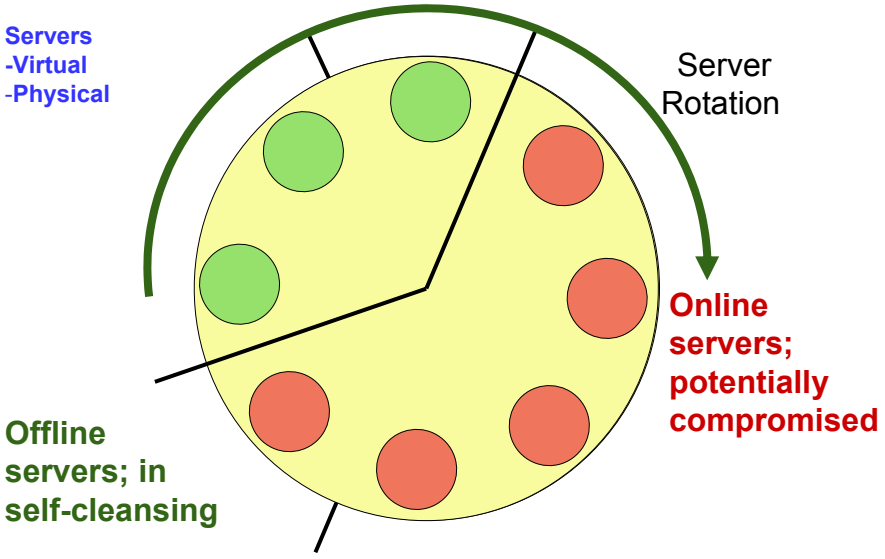
Server Rotations

Example: 5 online and 3 offline servers



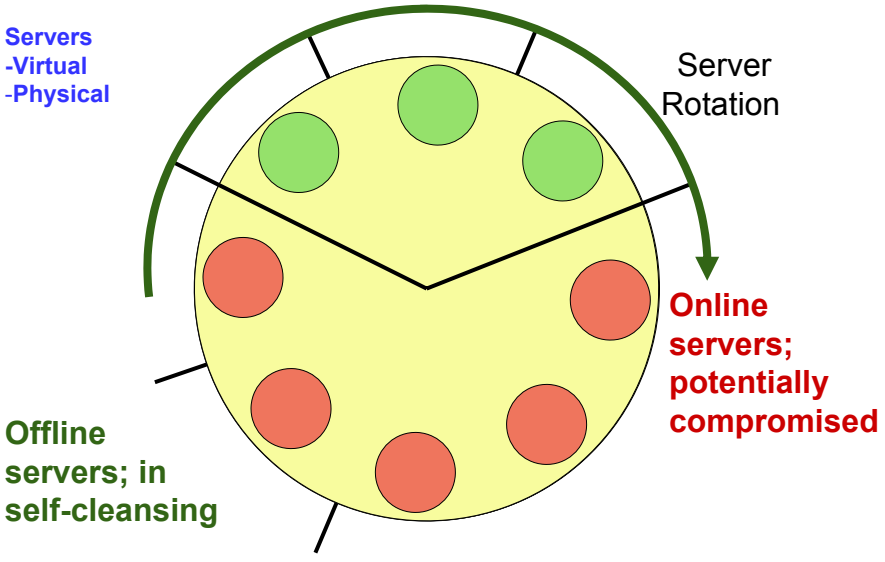
Server Rotations

Example: 5 online and 3 offline servers

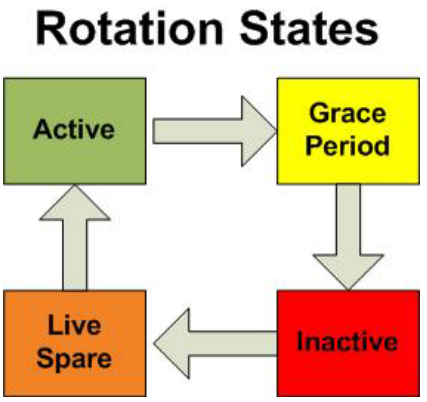


Server Rotations

Example: 5 online and 3 offline servers



Server State Transitions



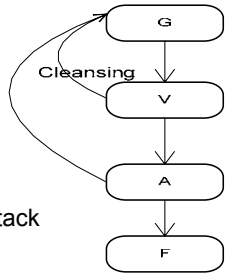
Outline

- SCIT Approach
- MTTSF Derivation
- MTTSF and SCIT Online Window
- Conclusion

SCIT Parameters

- Active window W_o: server accepts requests from the network
- Grace period W_g: server stops accepting new requests and fulfills outstanding requests in its queue.
- Exposure window: $W = W_o + W_g$.
- N_{total} : total nodes in the cluster.
- N_{total}, W, and the cleansing-time $T_{cleansing}$ are inter-related.

SCIT State Transition Diagram



G: Good
V: Vulnerable
A: Successful Attack
F: Failed

	G	V	A	F
G	0	1	0	0
V	$1-P_a$	0	P_a	0
A	P_c	0	0	$1-P_c$
F	0	0	0	1

- Simple diagram.
- Pa: probability of successful attack.
- Pc: probability of cleansing when in A.
- F: low chance of occurrence, but still possible:
  - Virtual machine and/or the host machine no longer responds to the Controller.
  - Controller itself fails due to a hardware fault.

MTTSF Computation Steps

- Following the methodology of Madan, et al – uses Semi-Markov Process model
  - Xa and Xt are absorbing and transient states
    - **Xa = {F} and Xt = {G, V, A}**
  - **q:** probabilities that process starts at each state in **Xt :**
    - **q = (1,0,0)**, since starts with state G.
  - **x:** visit count for each state in **Xt**.
  - **h**: mean sojourn times in each state
- Solve system of equations: **x = q + xP**
- Using solutions for **x**, compute $MTTSF_{scit}$ = **x . h**

MTTSF: Expression & Discussion

$$MTTSF_{SCIT} = \frac{\frac{h_0 + h_1}{P_a} + h_2}{(1 - P_c)}$$

- $P_a \downarrow \rightarrow MTTSF_{scit} \uparrow$
- $P_c \uparrow \rightarrow MTTSF_{scit} \uparrow$
- How to make $P_a \downarrow$ and $P_c \uparrow$?

Outline

- SCIT Approach
- MTTSF Derivation
- MTTSF and SCIT Online Window
- Conclusion

Relationship between P_a and W

- Modeling malicious attack arrivals:
  - Assumption: non-staged attacks
  - Attack arrivals ~ Poisson (λ)
- Then, inter-arrival time Y between attacks is exponential distribution:
  - $P(Y \leq W) = 1 - e^{-\lambda W}$
- $P(Y \leq W)$ = prob. that attacks occur in exposure window.
- Then:
  - $P_a \leq P(Y \leq W)$ {Only some attacks succeed.}
  - $\rightarrow P_a \leq 1 - e^{-\lambda W}$

Relationship between P_c and W

- Resident time of the attack modeled as a random variable Z for "service" time with rate μ.
- Assume Z exponential distribution:

  $P(Z > W) = e^{-\mu W}$
- System design is such that probability of moving out of state A due to the cleansing action is more than the probability that the service time is greater than W :
  - $P_c \geq P(Z > W)$
- On average system cannot "serve" more than the arriving attacks, then: $\mu \leq \lambda$.
- Then: $e^{-\mu W} \geq e^{-\lambda W}$.
- Hence: $P_c \geq e^{-\lambda W}$

MTTSF and W

- $W \downarrow \rightarrow (P_a \leq 1 - e^{-\lambda W}) \downarrow$
- $W \downarrow \rightarrow (P_c \geq e^{-\lambda W}) \uparrow$
- Then: $W \downarrow \rightarrow MTTSF_{scit} \uparrow$
- $MTTSF_{SCIT} \geq F(W)$, where F(W) is a decreasing function of W:

$$F(W) = \frac{\frac{h_0 + h_1}{(1 - e^{-\lambda W})} + h_2}{(1 - e^{-\lambda W})}$$

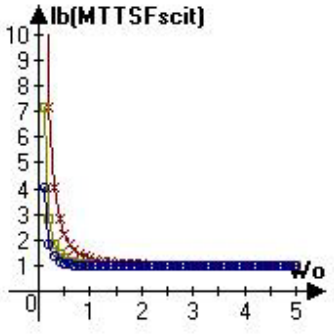
- Significance: engineer instance of SCIT architecture by tuning W in order to increase or decrease the value of $MTTSF_{SCIT}$.

MTTSF and Grace Period

- Grace period used by Controller to issue cleansing mode signal.
- $N_{outstanding}$: average # of outstanding requests in the queue when the server enters the grace period.
- Entire incoming traffic ~ Poisson(α).
- It is known: $\lambda = k.\alpha$, with $k \leq 1$.
- $N_{outstanding} \leq \alpha W_o$.
- S: service rate in terms of number of serviced requests per unit time:
  - $W_g = N_{outstanding} / S \leq (\alpha W_o) / S$
- Since $\alpha/S < 1$, estimate for grace period: $W_g < W_o$.
- Then: control $MTTSF_{SCIT}$ by online window W_o

Tuning Parameter: Exposure Time

- Trend of the lower bound $lb(MTTSF_{SCIT})$ in terms of W_o with example values of $h_i = (1/3)$ and $\lambda = 1$ (x), 2 (□), 3(○).



- Average Response Times for SCIT Persistent Web server tests [3].

Exp Time (minutes)	User Sessions	Avg. Response Time (secs)	STD Dev
2 m	50	6.16	0.07
2 m	100	6.24	0.01
2 m	**125**	**6.27**	**0.02**
3 m	50	6.10	0.04
3 m	100	6.15	0.02
3 m	**125**	**6.31**	**0.05**
4 m	50	6.08	0.04
4 m	100	6.15	0.02
4 m	**125**	**6.14**	**0.02**
No Rotation	50	6.03	0.01
No Rotation	100	6.03	0.00
No Rotation	**125**	**6.04**	**0.00**

SCIT Failure State

- Is state F really absorbing?
  - Compromise of Controller is very minimal due to the one-way data.
  - System automatically recovers back to the G state.
- Use Semi-Markov Process with embedded DTMC (Discrete-Time Markov Chain) to compute the steady-state *Availability* (state without security faults).
- Transition matrix **Q**

	G	V	A	F
G	0	1	0	0
V	$1-P_a$	0	P_a	0
A	P_c	0	0	$1-P_c$
F	1	0	0	0

Availability

- Solve the DTMC steady-state probabilities vector **y** = (y_0, y_1, y_2, y_3) for all states in {G, V, A, F}.
- Compute SMP stead-state probability π_F for state F:
  - $\pi_F = y_3 h_3 / \mathbf{y}.\mathbf{h}$, with **h** = ($h_0$, h_1, h_2, h_3) being extended to include the mean sojourn time h_3 for state F.
- *Availability* = $1 - \pi_F$

$$Availability = \frac{h_0 + h_1 + P_a h_2}{h_0 + h_1 + P_a h_2 + P_a(1 - P_c)h_3}$$

- *Availability* monotonically decreases with P_a but increases with P_c.
- Using the same line of reasoning and the assumption of Poisson attack arrival process as for $MTTSF_{SCIT}$ above, we can also conclude that decreasing the exposure window will increase *Availability* .

Conclusion

- SCIT servers have been built and tested
- Need a better way to choose parameters
- Paper has focused on evaluating the impact of key parameters on a security metric, and system availability.
- Important result: To make SCIT work well, we must increase Pc. Increases MTTSF and Availability.
- *We believe that the community should work towards defining and computing easily understood security metrics*

References

1. Yih Huang, David Arsenault, and Arun Sood. *Secure, "*Resilient Computing Clusters: Self-Cleansing Intrusion Tolerance with Hardware Enforced Security (SCIT/HES)". *The Second International Conference on Availability, Reliability, and Security, ARES 2007*.
2. Bharat B. Madan, Katerina Goseva-Popstojanova, Kalyanaraman Vaidyanathan, and Kishor S. Trivedi. "A Method for Modeling and Quantifying the Security Attributes of Intrusion Tolerant Systems". *Dependable systems and networks-performance and dependability symposium (DSN-PDS) 2002*.
3. Anantha K. Bangalore and Arun K Sood. "Securing Web Servers Using Self Cleansing Intrusion Tolerance (SCIT)". *DEPEND 2009*.

SCIT Publications + Contact Info

- SCIT technical publications
  - Links to media reports
  - Links to demo videos

http://cs.gmu.edu/~asood/scit

Workshop on Cyber Security and Global Affairs
Oxford University, UK

http://www.internationalcybercenter.org/workshops/cs-ga.2009

Questions?

Arun Sood

asood@gmu.edu
703.347.4494