Privacy-Preserving Distributed Event Correlation

Janak J. Parekh

Thesis Proposal

November 21, 2005

Overview of talk

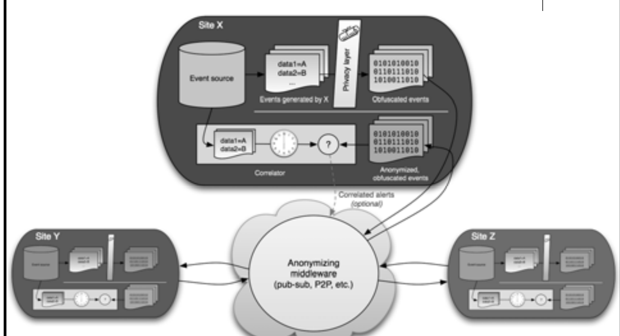
- Background
- Problem, Requirements
- Hypotheses, Solutions, Model
- Related Work
- Implementation, Feasibility, Next Steps
- Contributions, Accomplishments
- Schedule, Future Work

Background

Privacy-Preserving Distributed Event Correlation

- "Event correlation is the process of monitoring" in space and time, "what is happening on... systems in order to identify patterns of events that might signify attacks, intrusions, misuse or failure." [Kay03, edited]

Background

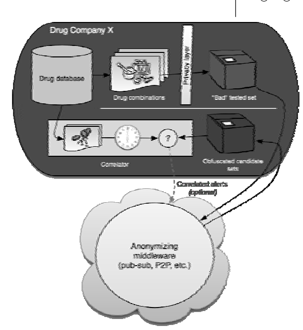


Problem

- "Intra-organizational" event approaches insufficient
  - Internet-scale applications need Internet-scale correlation, between organizations
  - Increase semantic richness via greater data collection
- Entities are reluctant to share information to competition, government, and/or malicious entities
  - Strategic: events may contain sensitive data/trade secrets
  - Compliance: government laws prevent information disclosure (e.g., HIPAA)
- Goal: balance information sharing and effectiveness of event correlation in a manner compatible with organizations' *privacy policies*

Problem: example

- Hypothetical drug testing scenario
  - Certain combinations of ingredients may have undesirable side-effects on people
  - Government requires that these combinations be disclosed for safety purposes
  - At the same time, drug manufacturer does not want to reveal trade secrets (e.g., which combinations they are testing)



What is "privacy"?

- Many different forms [EPIC05]
- *Source anonymity*: inability to trace the origin of events/identity of producer
  - Necessary between competitors, for example
  - "Anonymous tip hotline"
- *Data privacy*: avoid releasing confidential information
  - For example, internal data or networking information -- fundamental organizational structures
- With these two, we argue recipients cannot trace the source or information for relevant applications
- Not *time privacy*: for correlation, ordering is necessary

Requirements

- Support event source anonymity and data privacy
- Support event *corroboration*, i.e., common dataset intersection
- Support temporal constraints
- Support heterogeneous privacy policies, applications and data types
- Support authentication to the extent anonymity is not violated (e.g., group authentication)
- Near real-time performance (must be able to keep up with data streams)

Hypotheses

- The addition of *one-way data transformations* will enable effective corroboration despite organizational privacy-preserving requirements
- A *typed event-driven framework* supporting a range of one-way and two-way data structures enables matching heterogeneous privacy-preservation requirements

Solutions (I)

- Event source anonymity: leverage existing publish-subscribe systems with a trusted third-party (TTP) as authenticator/anonymizer
  - A general fully-decentralized architecture is extremely difficult [Douceur02], and is outside of thesis scope; use techniques like Onion routing [Goldschlag99] if necessary
  - Four levels of anonymity: non-anonymous, anonymous *but categorizable*, anonymous *but differentiable*, and fully anonymous
- Data privacy/anonymity via one-way data structures
  - Focus on <u>Bloom filters</u> [Bloom70]
  - Support multiple hash functions in the same data structure [Lincoln04] to defeat brute-force attacks
  - Can use less obfuscating data structures
  - Natively supports corroboration

Solutions (II)

- Event corroboration
  - Hashing solutions allow for set membership tests
  - Repeated hashing for aggregate/multiple type matching
- Temporal constraints
  - Rapid Bloom filter correlation via <u>**MRU** and **timestamp** Bloom filters</u>
  - Flexible timestamping mechanisms to support ordering
- Heterogeneous privacy policies, applications, types
  - Support correlation between heterogeneous messaging formats to allow for different privacy requirements
  - Motivate future development of privacy policy exchange language to automatically adapt data exchange and correlation based on what sites are willing to contribute

Model

1. Event *typing*
   - Support for hash-based data structures
   - Support for standard message exchange formats
   - Privacy-enabling metadata (base type, timestamp)
     - Versioning to support incremental datatypes, privacy evolution (cross-version compatibility/complex schemes outside scope)
2. Pluggable, event type-driven processing framework
   - *Heterogeneous* type-enabled correlation/corroboration modules
   - Translation facilities between formats
   - Correlation modules, legacy support for non-privacy correlators
3. Publish/subscribe event infrastructure
   - Leverage existing solutions (outside of thesis scope)
   - Support ordering, encryption, anonymization as needed

Related work: Event Correlation, Event Systems

- Temporal event correlation/aggregation supporting arbitrary event types
  - Rapide [Luckham96]: focus on software architecture simulation, monitoring
  - SMARTS InCharge/DECS [Yemini96]: primarily network, distributed application management
- Publish/subscribe content-based routing systems providing simple event filtering/covering
  - ELVIN [Segall00]: simple single-message predicate matching
  - Siena [Carzaniga00]: adds minimal support for sequence matching
  - Gryphon [Banavar99]: event stream "interpretation" to reduce transmission overhead

Related work: Distributed Intrusion Detection (DIDS)

- DIDS/CIDS: Distributed/Collaborative Intrusion Detection System, multiple networks and sensor(s) at each network
- GrIDS [Staniford96]: Graph hierarchy-based aggregation, with centralized monitoring server
- EMERALD [Porras97]: Distributed, component-based intrusion monitoring
- Quicksand [Kruegel02]: Completely decentralized, specification language to specify patterns
- Indra [Janakiraman03]: Uses "pub-sub-on-P2P" infrastructure
- DShield (Ullman, http://www.dshield.org): Volunteer DIDS
- DOMINO [Yegneswaran04]: Decentralized hierarchy with summary exchange; aggregate analysis of DShield logs

Related work: Privacy-Preserving Collaboration

- Corroboration most commonly implemented using set membership algorithms/tests
  - HotItem protocols [Kissner05]: Uses a Bloom filter implicitly; discusses theoretical capability to maintain "data" and "owner" privacy amongst malicious entities
- Hybrid approaches including hashing/set membership, randomized routing
  - [Lincoln04]: Hashing to scrub sensitive data, second key-based hash algorithm adds "noise" to prevent brute-force attacks
  - Friends Troubleshooting Network [Huang05]: build a recursive lookup P2P network that maintains anonymity; uses hashing, SMC, and random-walk routing for software diagnosis

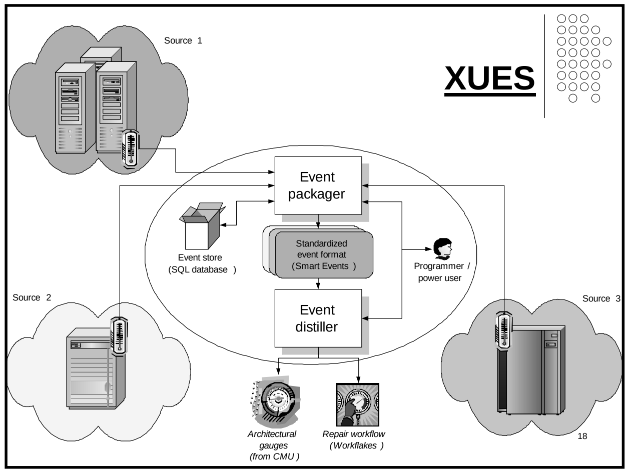
Related work: Other Privacy-Preserving Computation

- Statistical transformation: useful for larger data exchange where such "summaries" are accurate
  - PAYL [Wang05]: 1-gram and Zipf frequency distributions of packet content
- Databases and data mining
  - Statistical databases ([Agrawal00], [Lindell02): Aggregate statistics despite perturbation and individual restrictions
  - Privacy-preserving information sharing [Agrawal03]: Two-party equijoin, intersection, counts via commutative encryption
  - K-anonymity [Sweeney02]: Privacy via redundancy
  - Privacy-preserving BF-enabled queries [Bellovin04], secure indices [Bawa03, Goh04]
  - "Hippocratic databases" [Agrawal02]
- Secure multiparty communication [Yao82]
  - [Du01] proposes general transformation architecture, including intrusion detection information; too slow to handle near real-time alert streams

KX : XUES

- KX (Kinesthetics eXtreme): distributed application monitoring
  - Implemented model parts 2 and 3
  - Internet-scale (using Siena pub-sub architecture), but *not* privacy-preserving
- *Sensors* installed at each node to collect information
- XUES (XML Universal Event Service) processed events
  - Modules established *gauges* to measure application behavior from sequences of events over time
- *Behavioral models* drove system, defined gauges

XUES



KX/XUES Postmortem

- DARPA challenge problem: instrument and improve robustness of distributed GeoWorlds GIS/news visualization platform [Coutinho99]
  - Various services, e.g., noun phraser, would frequently “time out” and bring system down
  - Automated tool to instrument method calls in Java code, temporal correlation to detect service hanging
  - Workflow engine to restart services or load-balance automatically as necessary
- Other applications
  - Internet-scale deployment in joint work with TILab, instrumenting instant-message platform [Valetto03]
  - Used in AI2TV distance learning platform for bandwidth optimization and multi-viewer synchronization [Phung05]

WORMINATOR

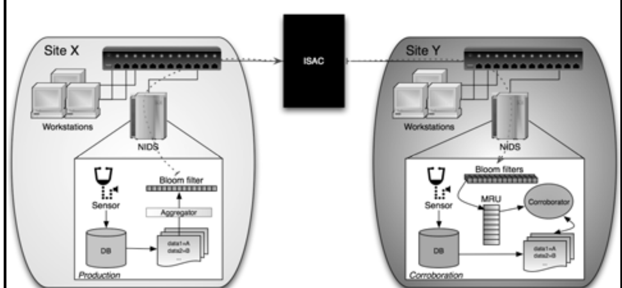
010XXX011XX101010011001110101XX110100001
COLLABORATIVE INTRUSION DETECTION

- Goal: correlate IDS IP-based alerts to detect common sources of scans and probes
  - Hypothesis: Using corroboration, detect not only worm spread, but *stealthy reconnaissance* for new attacks
  - Individual sensors produce voluminous amounts of alerts, making detection difficult
  - Commonality powerful indicator of intent: enable *profiling* of attacker behavior
- Not an IDS itself; a middleware layer that sits on top of existing misuse and anomaly detection sensors
  - Currently using Counterstorm Antura as an underlying sensor platform, supports very long-term scan detection

Worminator Implementation

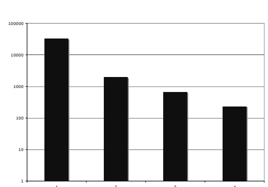
- Rebuilt XUES framework with privacy-preserving mechanisms for Internet-scale, cross-organization intrusion alert correlation
  - Implemented #1 from model
  - Current implementation leverages JBoss JMS publish/subscribe infrastructure, ISACs ideal trusted third party
  - Others in project experimenting with distributed P2P technologies
- Watchlist/warnlist model
  - Initially, goal is to find common source IPs and destination ports
  - Watchlists consisting of Bloom filters exchanged to prevent revealing sensitive network information
  - Warnlists containing corroborated sources may then be shared explicitly for proactive response mechanisms

Worminator



Worminator: Noise Reduction (9/12/05-10/10/05)

- In about one month, we acquired information on ~32,000 *new* IP addresses
- 1,924 IP addresses have scanned at least two of the sites;
- 659 sites have scanned at least 3;
- Only 232 have scanned 4
- Now deployed at 5 sites, more underway



What next?

- Longitudinal study of Worminator data
  - Demonstrate detection of stealthy scans not picked up by existing approaches (e.g., DShield)
  - Longitudes include *time*, *space*, and *target*
  - Motivate attacker profiling techniques (hand-crafted; automated profiling/modeling outside of thesis scope)
  - Preliminary study presented to ARO
- Evaluation of privacy-preserving methods
  - Optimize BFs: minimize size, brute-forceability
  - Information, temporal losses induced from BF use
  - Effectiveness of MRU and timestamp BF techniques
  - Use raw events as baseline

Expected contributions

- Deeper insight into modular architectures for cross-domain information sharing
- First steps towards a practical, deployed collaborative security system
- Development of fast BF corroboration data structures
- Evaluation of privacy-preserving mechanisms on corroboration
- Longitudinal study of stealthy scan behavior to evaluate CIDS

Accomplishments

- Publications: [Parekh05], [Locasto05], [Gross04], [Keromytis03], [Kaiser03], [Kaiser02], [Gross01]
- KX/XUES demoed, deployed in 3+ applications, Worminator currently deployed at 5+ sites (see http://worminator.cs.columbia.edu)
- Grant support, successful presentations and demos to DARPA, NSA, DHS, ARO
- Worminator technology licensed to Counterstorm, undergoing commercialization for DHS grant
- Patent application filed on aspects of Worminator work

Schedule

KX/XUES implemented, demonstrated	Done
Worminator: development, deployment	Done, testing/deploying
Worminator: longitudinal study	Initial study completed; writeup in Jan. '06
Privacy-preservation evaluation	March '06
Thesis distribution	July '06
Thesis defense	August '06

- Enables a broad variety of future applications…

Future applications

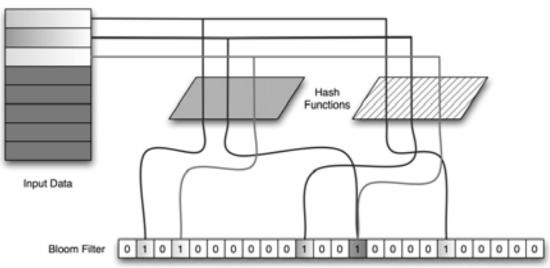
- Worminator "II": content alerts
  - Use payload anomaly detection
  - Verifies multiple-typing mechanisms alongside privacy mechanisms
  - Semantics and profiling of content alerts outside the scope of this thesis
- Posture-based [Knight02] aggregation/exchange policies
- Integrate privacy-preserving language and matching capabilities into Worminator
- "Application communities" peer-to-peer application monitoring
  - Idea: use application monoculture to bolster security and stability
  - Code/information sharing for application patch generation and distribution
  - DARPA proposal submitted, may extend this thesis work

Other future directions

- Other privacy mechanisms, e.g., solve malicious insider/watermarking problem?
- Evaluation of event distribution strategies
- Automated IDS attacker profiling
- Generalized event typing and versioning framework; possibly leverage FleXML
- Next-generation terminologies

Backup slides

Bloom Filter



Bloom Filter: Math

- Given an *m*-bit array, inserting *n* items using *k* hash functions yields a FP rate of approximately [Fan98]:

$$\left(1-e^{kn/m}\right)^{k}$$

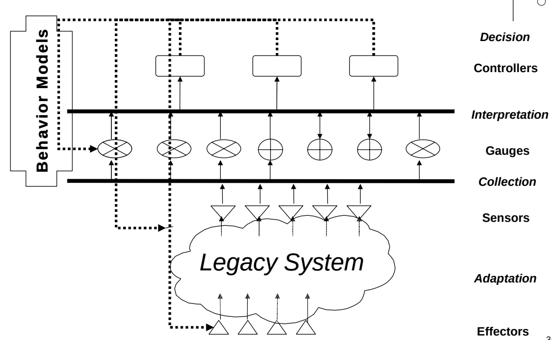
- To determine an optimal array length given a FP-rate *f* [Ceglowski04]:

$$m=\frac{-kn}{\ln\left(1-f^{1/k}\right)}$$

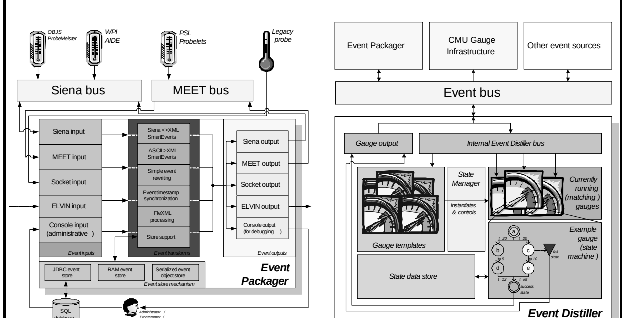
Bloom Filter Correlation

- Correlating against many collected Bloom filters is expensive; while Bloom filters can be ORed together, false positives increase as the bit array gradually becomes all 1s
- **MRU Bloom filter** supports aging by storing a timestamp for every bit, and supports *expiry*
- **Timestamp Bloom filter** supports temporal range queries by storing multiple timestamps for every bit
- BFs received from peers can be aggregated in as fast as O(*n*) time (MRU) or O(*n* lg *m*) time (timestamp); lookups are constant or logarithmic, respectively

KX: High-Level View



EP and ED, in detail



Worminator: Deployment

- Deployed sites
  - CUCS
  - CounterStorm, midtown Manhattan
  - Customer I, Washington D.C.
  - Florida Tech
  - Georgia Tech
- In process
  - Research and Education Network ISAC (Indiana, Syracuse)
  - USC
- Other prospective sites under discussion
  - CMU

Top Ports, 9/12-9/26

Port	# Alerts	Type
1434	34763	SQL
1026	6640	Messenger, backdoor
1027	2004	Messenger spam
135	980	NetBIOS
80	904	HTTP
1024	861	NetSpy
137	859	NetBIOS
3072	775	Backdoor/proxy?
4144	638	CompuServe!?
22	463	SSH

Interesting case

- The IP address 128.9.168.45 showed up on two sites within a few hours of initial operation back in March
- Resolves to http://ptr.isi.edu/
- We “caught” them!

DShield vs. Worminator

DShield	Worminator
Relies on *user-contributed alerts* from a wide variety of sensors, honeypots, etc.	Current focus is on a uniform, long-term NIDS
Geared towards groups that can disclose information (e.g., non-sensitive organizations)	Includes privacy-preserving policies to support critical infrastructure correlation
Ultimate focus is on reporting of suspect sources for end-user use	Ultimate focus is on stealthy behavior and profiling
Long-running project, lots of data, analysis (Yegneswaran et. al.)	Conceived in 2003, but getting critical infrastucture moving extremely difficult

Worminator vs. DShield

- Since March, we’ve identified 8,873 IPs detected at two or more sites
- Of these, we were able to query DShield about 7,261 records
- And of these, 3,880 *were not found by DShield*
- What *are* these guys doing?
- Future: gather more data from DShield, figure out opposite

Top 10 not in DShield but seen at 4 sites

Source IP	Num Alerts	Country	First Scan	Last Scan	Stealthiness
194.69.214.120	624	NO	3/16/05 10:23 AM	4/14/05 11:36 PM	0.000244404
220.189.245.70	540	CN	3/25/05 11:08 AM	9/14/05 6:15 AM	3.62E-05
61.152.117.17	539	CN	7/20/05 4:12 PM	9/14/05 1:53 AM	0.0001126
222.36.44.37	385	CN	9/15/05 4:34 AM	9/16/05 6:45 AM	0.00408354
61.152.91.231	373	CN	9/21/05 3:50 PM	9/22/05 12:19 PM	0.005058197
61.152.117.29	312	CN	9/13/05 2:26 PM	9/14/05 1:43 AM	0.007683187
61.178.136.101	253	CN	9/30/05 1:59 PM	10/1/05 8:30 AM	0.003794586
58.56.2.238	217	CN	10/17/05 9:14 AM	10/19/05 8:12 AM	0.001283238
60.195.7.82	212	JP	9/30/05 1:47 PM	10/1/05 6:49 PM	0.002027805
202.104.212.76	192	CN	9/9/05 3:42 PM	9/10/05 11:11 AM	0.002737296

Top 10 known ports amongst those sources

Count	Port	Service
8	514	syslog
5	7009	afs3-rmtsys
4	1911	mtp
4	6667	ircd
4	515	printer
3	1434	ms-sql-m
3	6010	x11-ssh-offset
3	73	netrjs-3
3	5680	canna
2	9	discard

How about stealthiness?

- Simple metric: # alerts / scan time

```
SELECT source_ip, MAX(last_scan_time) -
  MIN(first_scan_time) AS scan_length,
SUM(num_alerts),
SUM(num_alerts) / extract(EPOCH FROM
  (MAX(last_scan_time) - MIN(first_scan_time))) AS
  stealthiness
FROM worminator_watchlist_alerts
WHERE first_scan_time >= DATE '2005-09-12' AND
  first_scan_time <> last_scan_time
GROUP BY source_ip
HAVING SUM(num_alerts) > 1
ORDER BY stealthiness ASC
LIMIT 10
```


Stealthiness of IPs that have scanned 4 sites

Source IP	Scan length	# alerts	Stealthiness
60.18.168.112	14 days 01:57:17.095	67	5.51E-05
61.129.45.58	7 days 02:07:12.511	36	5.88E-05
213.172.46.218	3 days 02:16:42.516	17	6.36E-05
66.65.196.210	6 days 02:28:31.946	46	8.72E-05
219.136.53.213	5 days 05:19:32.322	41	9.09E-05
69.40.165.231	10 days 06:38:41.255	85	9.57E-05
61.145.112.71	4 days 21:32:39.754	42	9.93E-05
80.164.25.248	3 days 17:18:01.388	37	0.000115092
166.111.30.56	6 days 10:36:34.304	67	0.000120375
140.247.173.107	2 days 09:25:38.065	25	0.000120926

So what's 60.18.168.112?

- No reverse DNS (of course)
- In China
- Scanned 1434 at the two commercial entities (and *not* the academic ones)
- Scanned a whole ton of ephemeral ports on the academic ones (and, mostly, not the commercial ones)
- Misdirection?
- Botnet control only in .EDUs?
- Further research needed

Attacker Profile

- *Attacker Profile* = description of a set of attackers with similar malicious behavior
- Features that can be used to build the profile:
  - source IP
  - destination port
  - timestamps of the attacks or scans
  - content of the attacking packets or information about the content
  - type of the attacked sites (academic, commercial etc.)
  - maybe geography too

1. Stealthy Malicious ISP Profile (I)

- Distribute scanning load across large subnets or botnets to reduce individual node's activity and suspicion
- We were able to validate this hypothesis by examining s*ubnet aggregation*
- Several particular results stood out…

1. Stealthy Malicious ISP Profile (II)

- Scanning the same port from "almost" consecutive IPs
- Seen at two academic sites and one commercial site
- Infected cable modems wouldn't have this kind of distribution
- Country: US

IPs	Port
66.194.6.2	80
66.194.6.67	80
66.194.6.68	80
66.194.6.70	80
66.194.6.71	80
66.194.6.72	80
66.194.6.73	80
66.194.6.74	80
66.194.6.75	80
66.194.6.76	80
66.194.6.77	80
66.194.6.78	80
66.194.6.79	80
66.194.6.80	80
66.194.6.81	80
66.194.6.83	80

1. Stealthy Malicious ISP Profile (III)

- Scanning subnets were observed at one site, but considering the number of hosts it's unlikely it's targeting *only* that site
- Scaling up will enable us to better detect the breadth of these scanning attempts
- US? *TT!?*

Subnet	Country	Number of scanning IPs
209.94.161.0/24	US	254
209.94.194.0/24	TT	254
209.94.210.0/24	TT	254
209.94.214.0/24	TT	254
209.94.219.0/24	TT	254
209.94.199.0/24	TT	253
209.94.215.0/24	TT	251
209.94.208.0/24	TT	243
209.94.212.0/24	TT	241
209.94.134.0/24	US	240

1. Stealthy Malicious ISP Profile (IV)

```
OrgName:    Wood County Telephone Company
OrgID:      WCTC
Address:    440 E Grand Avenue
City:       Wisconsin Rapids
StateProv:  WI
PostalCode: 54494
Country:    US

ReferralServer: rwhois://lombardi.wctc.net:4321

NetRange:   209.94.160.0 - 209.94.191.255
CIDR:       209.94.160.0/19
NetName:    WCTC97
NetHandle:  NET-209-94-160-0-1
Parent:     NET-209-0-0-0-0
NetType:    Direct Allocation
…
```

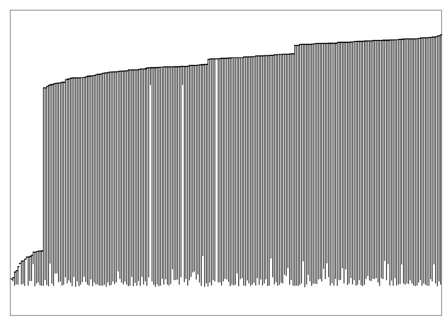

1. Stealthy Malicious ISP Profile (V)

- Scanning subnets' activities

IPs	Ports	First scan time
209.94.161.1	135 139 445	2005-03-21 21:32:22.76
209.94.161.2	80 135 139 445 1025 1433 2745 3127 6129	2005-03-18 21:48:25.632
209.94.161.3	80 135 139 445 1025 2745 3127 6129	2005-03-16 01:56:40.714
209.94.161.4	135 445 1025 2745	2005-03-16 00:15:44.899
209.94.161.5	135 139 445 1025 2745 3127 6129	2005-03-15 21:07:33.142
209.94.161.6	135 445	2005-03-20 05:05:46.513
209.94.161.7	80 135 139 445 1025 2745 3127 6129	2005-04-09 14:15:49.925
209.94.161.8	135 445 2745	2005-03-23 15:37:46.893
209.94.161.9	135 445 1025 2745	2005-03-15 21:33:50.763
209.94.161.10	135 445	2005-03-16 03:28:44.053

1. Stealthy Malicious ISP Profile (VI), based on timestamps

Start and end times of 209.94.161.*



2. Fixed Schedule Profile (I)

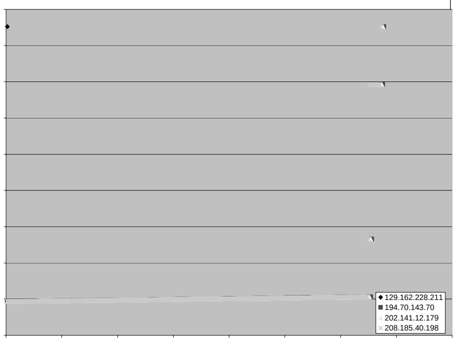
- IPs that are scanning exactly in the same time interval on the same host

First scan	Last scan	IP	Country	DShield
2005-10-01 14:05:38.761	2005-10-02 18:50:23.04	129.162.228.211	US	NO
		202.141.12.179	AU	YES
		194.70.143.50	GB	NO
		208.185.40.198	US	NO
2005-03-15 21:48:17.897	2005-03-16 22:25:22.833	218.14.157.104	CN	NO
		218.14.157.80	CN	NO

2. Fixed Schedule Profile (II)

IP	Number of alerts generated	Behavior	Top ports	
129.162.228.211	341	Scanning the same 210 etc ports on the same academic site	7002	afs3-prserver
202.141.12.179	338		749	kerberos-adm
194.70.143.50	339		513	login
208.185.40.198	336		347	fatserv
			107	rtelnet
			1701	12tp
218.14.157.104	3	Scanning the same commercial site		
218.14.157.80	3			

2. Fixed Schedule Profile (III), timestamps for all 4 sites



3. Worm Profile
(based on scanning information)

- IPs hitting 4 sites on the same port
- The timestamps of the attacks have to be close to each other
- e.g. for 210.103.67.65 the timestamps on each site were:
  - 2005-09-18 20:13:06
  - 2005-09-18 20:14:49
  - 2005-09-18 20:22:48
  - 2005-09-18 23:30:34

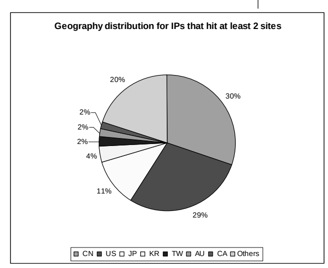
IPs	Port
61.142.246.194	1434
61.145.227.5	1434
61.153.143.164	1434
61.183.13.183	1434
193.165.168.42	80
200.81.220.250	1434
202.99.160.209	1434
202.105.237.2	1434
210.74.224.79	1434
210.103.67.65	80
216.74.57.104	1434
218.25.10.87	1434

4. Content Profile

- The profile describes the set of the attackers that generate the same anomalous content
- Detecting the distributed subnets
  - Fellow researchers at GA Tech working on the latest botnet detection techniques
  - Integration of *payload anomaly detection* into Worminator enables *content* profiling, without dependency on IP address distributions
- Correlating with the worm profiling

5. Geography Profile

- There are particular countries with a predominant malicious behavior
- Proportional to technological density?



Next step: Scale up

- Once we have more sites online, we can glean more data about the stealthy, subtle scanners across different classes of networks
- Utilize PAYL to determine what the attack payload is
- Use content and network modeling to build a *profile* of the attacker, hopefully before the attack itself
- Worminator serves as a good underlying platform, and our research merits further development

Worminator demonstration

- Centralized management overview of all sites
- Site drill down and raw alerts
- Top 50 alerts by site
  - Drill down on source IP incl. WHOIS and Trace Route
  - Common port scans
  - Length of scan time
- Correlation of multiple sites
  - Top 50 alerts
- https://worminator.cs.columbia.edu/