GLOBAL
CYBER SECURITY
CENTER

DNS Metric Use Cases

Authors: E. Casalicchio, D. Conrad, Joao Damas, S. Di Blasi, I. Nai Fovino

Document version:	***0.7***
Authors:	***E. Casalicchio, D. Conrad, Joao Damas, S. Di Blasi, I. Nai Fovino***
Content:	***DNS vulnerabilities metrics***
Related project:	***Mensa***
Date:	***19/05/2011***
Comments:	**---**

Table of contents

1 Introduction

This document proposes a set of metrics that may be used to evaluate the health of the DNS by measuring the DNS along three dimensions, namely Vulnerabilities, Security, and Resiliency.

The most common DNS vulnerabilities, present in many threats scenarios such as those discussed in [3] are: Cache Poisoning, Distributed Denial of Service (DDoS), Response Modification, Route Injection, and Origination Modification. Such hazards, and many more, can be classified into five main threats categories as reported in [2]: Data corruption (Repository corruption, System corruption and Protocol issues), Denial of Service and Privacy Violation. In this document, Vulnerability metrics are organized along five categories that match the main DNS vulnerabilities mentioned above.

Metrics characterizing security of the DNS, defined as the ability of the DNS to limit or protect itself from malicious activity (e.g. unauthorized system access, fraudulent representation of identity, and interception of communications), have yet to be defined. While the deployment and use of DNSSEC, which allows for DNS data integrity to be assured, is growing, how this deployment impacts the security level of the DNS is as yet unknown. In this document we propose a set of metrics that, taken together, can contribute to the evaluation of DNS security readiness with respect to a possible set of attack scenarios.

DNS Resiliency is defined to be the ability of the DNS to effectively respond and recover to a known, desired, and safe state when disruption occurs (e.g., response and recovery after a distributed denial of service attack). Resiliency is viewed by users as availability and viewed by providers as a combination of detection, response, resistance and recovery processes that increase overall confidence in relying on and investing in the Internet over the long-term. DNS Resiliency can be also described as the ability of the DNS to provide and maintain an acceptable level of service in face of faults and challenges to normal operations. In this document we use a set of metrics aimed at measuring the Resiliency of a generic ICT system and apply those metrics to the Resiliency of the DNS.

It is important to stress the metrics described in this document represent an initial effort, more intended to initiate discussion than to be considered definitive.

1.1. Metric Categories

Repository Corruption. A repository is a central place in which data is stored and maintained. For the DNS this is the authoritative source of the zone information. Depending on how the DNS zone and systems serving the zone are operated this can be the raw on-disk zone files or an administrative database. Examples of Repository Corruption would include unauthorized authoritative or cache database manipulation.

System Corruption. The authenticity of DNS responses is fully dependent on the trust of the whole chain of systems in the (relevant part of the) DNS tree and the systems that traverse that chain. Generally (and by design) not all of these systems are under control of the same entity. This makes it difficult if not impossible for the owner of the DNS data to fully ensure data authenticity to the client. Examples of system corruption risks are Response and Origination Modification threats.

Protocol Issues. This category of vulnerabilities deals with incidents further down the DNS response tree viewed from the perspective of the authoritative servers in which the DNS protocol itself (or protocols used in support of the DNS) is exploited. Examples of protocol issues are cache poisoning, route injections and man-in-the-middle threats

Denial of Service attack refers to a type of attack that renders the service unusable for legitimate users. These attacks are either aimed at a specific service (like the DNS) or aimed wider to a whole part of the network (Internet). We may thus distinguish between DoS attacks against DNS servers and those against Network Infrastructure. Examples of DoS attacks are network traffic flooding attacks that overwhelm bandwidth, routers, or network interfaces and communication fiber cuts in which all communication bandwidth to a DNS facility is disabled.

Privacy Violation. Some problems with DNS are not so much a security issue in the sense that they change data, rather they are privacy related issues as they allow attackers to get insight into how the DNS or more generally, the Internet is used. Cache snooping and NSEC walking are example of Privacy Violations.

Resilience is the ability of the DNS to effectively respond and recover to a known, desired, and safe state when disrupted. Examples of resilience would include recovery from Denial of Service and restoration of correct data after repository corruption.

Security is the ability of the DNS to limit or protect itself from malicious activity (e.g. unauthorized system access, fraudulent representation of identity, and interception of communications). Security provides increased confidence in the DNS. Examples of Security would include the deployment of DNSSEC and use of TSIG to protect stub resolver to caching resolver communications.

1.2. Metrics description

Each proposed metric is described with a template that contains:

- **Measure**: the name of the metric
- **Method**: how the metric can be obtained
- **Metric:** the value of the metric
- **Use:** how the metric can be used
- **Discussion:** Consideration about the metric itself, the data needed to compute the metric, the measurement point to collect the appropriate data, limitations.

2 Vulnerability Metrics

This section provides a description of proposed metrics for DNS vulnerabilities. Following each table is a discussion of the proposed metric.

2.1. Repository Corruption

Measure	Data Staleness
Method	Comparing SOA serial numbers among all authoritative servers
Metric	Percentage of differing SOA serial numbers across all authoritative servers numbers over a time period
Use	A non-zero metric over a long time period would suggest an inability for secondary name servers to keep up with the primary.

Discussion

The Data Staleness metric tries to establish whether some secondary servers are unable to process zone updates occurring at the primary. This could occur when a secondary has limited and/or unstable connectivity to the primary.

Implementation of this metric would entail first obtaining a list of all authoritative servers for a zone, then periodically polling those servers for SOA changes. Obtaining the list of authoritative servers is possible by querying the parent zone for the NS records. If cooperation of the operator of the primary/master is available, the polling could be triggered by DNS "NOTIFY" messages. Upon detecting a change, the entire list of name servers could be polled repeatedly over a short interval to determine the time period until the zone fully synchronizes across all name servers.

Measure	Data Staleness Duration
Method	Comparing SOA numbers among all authoritative servers
Metric	Number of seconds until all SOA serial numbers are the same
Use	The maximum number of seconds until all sequence numbers are synchronized can be used to establish the maximum rate in which zone data can be updated.

Discussion

The Data Staleness Duration can measure the time needed for zone convergence across different instances of authoritative name servers.

Implementation of this metric is similar to measuring Data Staleness previously described, requiring essentially the same polling mechanism. In this case, the value of the SOAs returned by the authoritative servers is irrelevant, what matters is how long it takes for the SOAs to re-synchronize.

Measure	Zone drift/zone thrash [9]
Method	Check values for SOA fields: Refresh value, Retry value, Expire value, Minimum TTL, zone file change rate.
Metric	Probability of incurring in zone drift and zone thrash status
Use	The refresh value in the zone SOA RR should be chosen in accordance with average zone file changes frequency. If the zone is signed (DNSSEC enabled), the refresh value should be less than the RRSIG validity period. Retry, Expire and Minimum TTL value can vary according to specific service requirements.

Discussion

SOA RR field values are important in that they should be set in accordance with observed zone file change frequency, otherwise systems might incur in either zone drift status, when Refresh and Retry fields are set too high and zone file changes frequently, causing stale zone data at slave name servers, or zone thrash status, when Refresh and Retry fields are too low with respect to zone file changes, causing frequent zone transfers, thus imposing a workload burden over both master and slave name servers. This might lead to denial of service scenarios.

Implementation of this metric would require polling the zone SOA serial number at the primary/master for the number of zone updates within the zone refresh period.

Measure NS Parent/Child Data Coherence

Method Comparing the responses to NS queries to the parent zone with the responses to NS queries among all authoritative servers for the zone within one serial number.

Metric Percentage of differences

Use A non-zero percentage of differences could represent misconfiguration among the authoritative servers.

Discussion

Typically, a properly configured zone would have identical name servers between the parent and the child. It should be noted that intentional differences can exist when one or more name servers for the zone are in the process of being changed.

Implementation of this metric would require querying the servers for the parent zone for the name servers of the child zone, then querying all the listed name servers for the child zone for their view of the name servers for the zone.

A particularly bad case of incoherency occurs when the parent zone contains delegation records that lead to servers which are not configured to serve the intended child zone (lame delegations) as these lead the DNS resolution process down a dead end from which it must recover by backtracking and trying new servers from a now reduced pool of available servers.

Measure Glue inconsistencies

Method Comparing the glue responses between parent and authoritative servers for the zone.

Metric Percentage of differences

Use A non-zero percentage of differences could represent misconfiguration among the authoritative servers.

Discussion

The glue records associated with name servers should be consistent among all authoritative name servers and the parent.

A further complication of this scenario involves the detection of problems in sibling glue a situation in which cross references between nameservers for different zones are such that each depends on the other in a situation that can only be escaped from if both zones contain correct glue.

Implementation of this metric would be to query for the address records for each of the name servers listed for the zone, both at the parent and at all of the authoritative servers for the zone itself.

Measure	Zone inconsistencies
Method	Comparing the responses to queries among all authoritative servers for various RRs within one serial number.
Metric	Percentage of differences
Use	A non-zero percentage of differences could represent misconfiguration among the authoritative servers.

Discussion

Within one serial number, resource records in zone data should be consistent among all authoritative servers. It is known that name servers for some content distribution networks provide different answers depending on the query source IP address (or other identifiers).

Implementation of this metric would require access to the zone contents from each of the authoritative servers. After the authoritative servers synchronize their serial numbers, a zone transfer can be requested from each authoritative server and the results compared.

2.2. System Corruption

Measure	NXDOMAIN Redirection
Method	Comparing data returned by a query to a caching server for a non-existent name vs. the data returned by querying the authoritative servers for the same non-existent name directly.
Metric	Binary
Use	A true response would indicate redirection is in use.

Discussion

At the moment we are focusing just on NXDOMAIN field. NXDOMAIN modification might be at that point generalized as DNS Response Modification.

Implementation of this metric would require querying a caching server with a name known not to exist. Redirection would be in use if a positive response is returned.

A further extension to this metric would include NODATA response redirection, in particular to DNS queries regarding A, AAAA records.

Additionally, authoritative server can also perform NXDOMAIN redirection, upstream from the DNS resolver. A prime example of this in the real world was VeriSign's sitefinder service.

Measure	NXDOMAIN detection rate [11]
Method	Detecting the frequency at which NXDOMAIN replies occur
Metric	NXDOMAIN replies rate
Use	An abnormally detection rate might be an indicator of a botnet exploiting Dynamic DNS services to setup communication with C&C servers.

Discussion

Abnormally recurring NXDOMAIN replies might be a likely indicator of botnet activity over Dynamic DNS services. Deep analysis should be established in order to filter out false positives.

Alternatively this could be the result of ISP policy, redirecting NXDOMAIN (or NODATA) responses to some ISP specified service, with or without user content.

Implementation of this metric would likely require access to the query log of the caching resolver(s) or access to the query/response path of the resolver(s). Observing the number of NXDOMAINs over a period of time would establish a baseline. Significant increase in NXDOMAIN over that baseline would suggest unanticipated traffic and could trigger further analysis.

Measure	Data availability
Method	Comparing data returned by a query to a caching server for a existent data for a name vs. the data returned by querying the authoritative servers for the same non-existent name directly.
Metric	Binary
Use	A true response would indicate type filtering is in use.

Discussion

This could be the outcome of non-compliant behavior from middleboxes such as firewalls that have a limited understanding of the DNS protocol.

This metric measures availability of specific data types at the point of measurement. A typical case may be the (un)availability of record types not in common use at this time, such as AAAA records or any of the DNSSEC record types.

It may also be perceived as the reception of truncated responses where part of the data is missing because of narrow and outdate interpretations of payload limitation in the DNS protocol.

Measure	Domain blocking
Method	Comparing data returned by a query to a caching server for a existent data for a name vs. the data returned by querying the authoritative servers for the same non-existent name directly.
Metric	Binary
Use	A true response would indicate type filtering is in use.

Discussion

This metric is can be seen as the inverse of NXDOMAIN redirection whereby an existent domain is filtered or redirected by virtue of local policy implemented at the DNS resolver.

Policies that could result in this effect vary from anti-spam measures (using for instance the Response Policy Zone (RPZ) in BIND) to official government or corporate filtering policies to prevent access to certain domains.

On occasion direct access to authoritative servers from within a network may not be possible.

2.3. Protocol Issues

Measure	Cache Poisoning
Method	Comparing contents of caches vs. authoritative data
Metric	Percentage of differences
Use	A non-zero percentage would suggest the cache was being poisoned

Discussion

The *Percentage of differences* metric is evaluated by the cache operator and is valid at a give time instant. If N is the number of elements in the cache and $S(t)=\{t\_i \mid i=1..N\}$ is the set of are the remaining validity time for the records in the cache at time *t*, the "Percentage of difference" at time *t* is valid for $\Delta T=t+min(S(t))$.

Care must be taken to distinguish between cache poisoning and the presence of data in the cache of data that was valid in the recent past and is in the cache because its

TTL has not yet decremented to 0. A high rate of stale cached data may be an indicator of fast flux DNS in use, usually with malicious intent.

Within a resolver operator's administrative domain, implementation of this metric could be done via periodic dumping of the cache for particular zone contents after the cache had been primed for those contents. The results from that cache dump could then be compared with the known correct values. Alternatively (or outside the resolver operator's domain), queries for zone contents could be performed against open resolvers, comparing the response to known correct data.

Measure	Cache Poisoning Rate
Method	Comparing contents of caches vs. authoritative data
Metric	Poisoning rate = Number of times over a time period T data in the cache are different from authoritative data
Use	A non-zero value would suggest the cache was being poisoned in T. Higher the value higher the cache vulnerability

Discussion

The *poisoning rate* metric is evaluated by the cache operator and gives a picture of the cache vulnerability against poisoning attacks.

Implementation of this metric would be similar to implementation of the Cache Poisoning metric previously discussed, merely repeated over time period T.

Measure	Cache Poisoning Probability
Method	Evaluating the probability that the server provides a poisoned resource record
Metric	Probability that the server provides a poisoned resource record (*P_a*) computed as suggested in [1]
Use	A non zero probability value means the cache is under attack

Discussion

This metric could be valuable to understand what is the probability that a name server under attack provide a poisoned record and therefore that is the probability that a client is redirected to a resource controlled by the attacker.

Another interesting metric, used in the definition and computation of *P_a* is the *time taken for the poison to propagate*.

Implementation of this metric would be similar to the implementation of Cache Poisoning Rate, however looking at the percentage of queries that are returned poisoned versus the percentage that are returned correct.

Measure	Cache Poisoning Propagation
Method	Evaluating the time taken for the poison to propagate
Metric	Time to propagate, computed as suggested in [1]
Use	Lower the value higher the effectiveness of the attack

Discussion

To compute the *time to propagate* metric could be evaluated operating information sharing among DNS cache operators.

Implementation of this metric would be similar to the implementation of Cache Poisoning, however instead of looking at a single cache for poisoned records, a number of caches would be analyzed for poisoning.

Measure	DNS spoofing
Method	Evaluating the probability of spoofing a resolver
Metric	Probability of being spoofed (*P_s*) and probability of being spoofed over a time period T (*P_cs*) as defined in RFC5452 [2]
Use	*P_s* or *P_cs* give a measure of the chance to spoof a DNS resolver

Discussion

Being spoofed could mean to be used as an amplified of DoS attack.

The *P_s* and *P_cs* metrics could be evaluated at the resolver. The evaluation of the metric require to know parameters such as Number of ports available for use, number of authoritative name servers for a domain, average response time of each authoritative server.

Implementation of this metric would be to issue queries with a spoof source address to IP addresses suspected as operating as a DNS resolver. If the spoofed source address receives the response to the query, it would indicate the IP address to which the query was sent was an open resolver.

Measure	Zone Transfer failure
Method	Observing the number of zone transfer requests from the same source
Metric	Number of failed zone transfer operations
Use	A high number of zone transfer failures can express the existence of DoS attack on primary and secondary authoritative name servers, thus preventing smooth zone transfer operations.

Discussion

Zone transfer failures are typically logged in the system logs of both the primary and secondary servers.

Implementation of this metric would require either access to the query logs of the authoritative servers or monitoring of the query path to those servers.

Measure	Zone Transfer protection
Method	Observing if a slave server performs verification of the identity of the master server before accepting a zone transfer
Metric	Detect usage of TSIG or IP based ACLs
Use	Unprotected zone transfers could result in spoofed data being inserted into the DNS server.

Discussion

While zone transfers are typically performed over TCP, which provides better protection against spoofing that UDP, additional protection is necessary to ensure the authenticity of communications between master and slave servers when performing zone transfers.

2.4. Denial of Service

Measure	Variation of DNS Requests per Second
Method	Comparing the increase of the number of DNS requests per second with respect to the average number of DNS requests per second.
Metric	Variation of the requests number per second
Use	A rapid increase of the DNS requests would indicate the begin of a possible DOS attack

Discussion

This metric can give an indication of the DNS traffic at a DNS server. An unusually high variation of requests/second may represent an anomaly that may indicate a flooding DOS attempt is underway. In other words, could be taken as an index of speed/aggressiveness of a local DNS DOS attack

This metric could be implemented by sending repeated queries to establish a baseline round-trip-time (RTT) for a particular server and then probing that server to determine if RTT increases significantly. In order to avoid false positives to to routing system changes, the probes should be done in conjunction with checks of the routing path (e.g., traceroute).

Measure	Variation of DNS Request type distribution in time
Method	Comparing the variation of the distribution of DNS requests types per unit of time with respect to the steady state pattern
Metric	Variation of the distribution of request types per second
Use	A significant variation in the distribution of request types could indicate the begin of a possible DOS attack on the DNS or other systems

Discussion

Observations of past incidents involving compromised systems indicate that some of these systems, either due to design or defect, alter the pattern of request types reaching a name server and make this pattern an indication of potentially malicious behaviour.

Measure	Incoming Bandwidth Consumption

Method Measuring the amount of IP traffic directed to a target DNS server

Metric Percentage of available bandwidth.

Use Provides a rough measure that might indicate how close a DNS server is to its bandwidth saturation point.

Discussion

This metric is, generally speaking, a possible indicator for the Resilience of the DNS server. If the average bandwidth consumption is near to the saturation point, the DNS server will be less resilient to bandwidth consumption DOS attacks

Implementation of this metric would require monitoring of all bandwidth in to a DNS server, ideally measured from bottleneck points (i.e., not measuring the local LAN bandwidth but instead measuring the narrowest bandwidth connecting the DNS server to the outside world).

Measure Incoming traffic variation

Method Measuring variation of the amount of IP traffic directed to a target DNS server

Metric ΔMbit/sec

Use Provides a first derivative measure of the network traffic against the local DNS server.

Discussion

A rapid variation in the amount of the traffic may represent an anomaly worthy of investigation as it may suggest the beginning of an attack.

Implementation of this metric would require monitoring incoming traffic into a DNS server, establishing a baseline, and triggering an alert when there is a significant increase over that baseline.

Measure Geographical DOS Effectiveness

Method Comparing the DNS request timeouts for the same pool of requests using clients geographically spread in different continents.

Metric For each geographical area, (number of timeouts)/(number of requests).

Use It provides a rough measure of the global quality of the DNS service

Discussion

In case of large scale DOS against a particular part of the DNS this metric could be used to measure the impact extension of the attack

Implementing this metric would require having probes geographically distributed on a continental basis and using those probes to query servers, measuring the timeout rate at each probe.

Measure	Rate of repeated queries
Method	Analyze the traffic logs to quantify the number or repeated queries
Metric	Number of repeated queries
Use	A too high number of repeated queries might indicate a misconfiguration of firewalls and routers between the DNS server and the client (e.g. the firewall allows the requests, but cut systematically the answers)

Discussion

It might indicate a DOS in act against the FW protecting the client or a DOS against the DNS Server, unable to answer to the queries.

Measure	Traffic Tolerance
Method	Measuring the increase of delay with respect to the increase of traffic
Metric	Δ traffic/ Δ transaction delay
Use	It provides a measure of the relation between the network congestion and the resulting introduced transaction delay

Discussion

The measure can be used to quantify the in a what-if fashion the impact of a DDOS based on bandwidth consumption against a target DNS server.

Measure	Data Pollution
Method	Measure the aggregate volume and rate of bogus queries such as A-for-A, private address spaces and bogus TLD.
Metric	Percentage of queries carrying DNS pollution
Use	It provides a measure of the relation between the network congestion and the resulting introduced transaction delay

Discussion

This metric can provide useful hints for implementing advanced preventive traffic filters and prevent bogus queries processing; DNS-related software vendors can also take advantage of this information in order to fix reported bugs in order to reduce systemic pollution.

Measure	Open recursion/traffic spoofing
Method	Measures availability of open recursion from networks other than the ones directly served by a DNS resolver
Metric	Binary: availability or not of recursion services from external networks
Use	Provides a view of the potential of the resolver to be used as a reflector of spoofed DNS traffic as a means of generating traffic against other targets

Discussion

This metric provides useful hints for assessing the potential use of DNS recursive servers as a means of generating traffic (potentially DoS traffic) towards other targets**Errore. L'origine riferimento non è stata trovata.**.

2.5. Information exposure

Measure	Zone Walkability
Method	Using NSEC records to 'walk the zone'
Metric	Binary
Use	A true response would indicate the zone is walkable and thus may permit privacy violations.

Discussion

Use of NSEC implies the ability to traverse the zone, exposing all zone data.

Measure	Name Exposure
Method	Searching for unpublished zone data via tools such as Google.

Metric	Percentage of names in zone discoverable
Use	A higher percentage of names found would suggest zone data was being harvested in some manner.

Discussion

Determination of zone data that is unpublished could be done via a brute force approach (e.g., querying for all LDH strings less than some length or by using a dictionary of likely DNS labels derived from a source such as the ISC DNS Survey), however this may be taken as an attack. Another approach would be to use DITL-type data and extract query names, however this may violate the terms of use of DITL data. The approach least likely to have negative repercussions may be to ask zone owners for copies of their zones.

Measure	RR types information leakage [9]
Method	Quality check on RR types HINFO (Host Information), Responsible Person (RP), Location (LOC), (TXT) Record
Metric	Probability of having relevant information disclosed useful to attacker
Use	DNS administrator should take care when including HINFO, RP, LOC, TXT or other RR types that could divulge information that would be useful to an attacker.

Discussion

DNS deployment best practices suggest these RR types should be avoided if possible in publicly available zones, recommending their use only o the internal sides of split DNS scenarios.

2.6. DNSSEC deployment/configuration

Measure	Keyset availability [10]
Method	Iterated query for zone keyset (KSK/ZSK)
Metric	Percentage of key-set availability to a set of resolvers in a time range
Use	This measures the degree the system can provide DNSSEC keyset data to requesting resolvers.

Discussion

The availability of keysets involves not only the publication of the keys but also server configuration aspects such as ensuring the server supports an EDNS buffer size big enough to enable transport of the keys over UDP, without forcing fallback to TCP.

Measure Verifiability [10]

Method $1- ((|Ta|-1/)|Zs|)$, where $|Ta|$ is number of trust anchors to cover all secure zones and $|Zs|$ is the number of secure zones.

Metric v in [0…1]; v=1 indicates there is only one trust anchor needed to complete the verification chain (best case); v<1 indicates the resolver needs more than just one trust anchor, and it can decrease according to the unsecured zone density.

Use This measures whether the end-system can cryptographically verify the data it receives on the basis of the targeted secure zones and the minimum number of trust anchors needed to cover all the secure zones.

Discussion

As resolvers must be able to verify keysets, they have to be initially provided with a set of key from trusted zones; not all zones are secure and there are gaps in the authentication chain and these gaps must be filled by adding additional trust anchors.

In the worst case, there could be no authentication chains and resolvers would need to be configured with a trust anchor corresponding to each secure zone.

In the best case, there would need only one trust anchor to cover all the authentication chain.

Measure Availability of DNSSEC verified data to user applications

Method Request DNSSEC data from recursive server

Metric Detect setting of AD bit in DNS response from recursive server.

Use This measures whether the recursive is cryptographically verifying the data it receives on the basis of the targeted secure zones and the minimum number of trust anchors needed to cover all the secure zones.

Discussion

In the early stages of DNSSEC deployment most verification will be performed at the recursive servers. These signal success in verification by setting the AD bit in the DNS response to the end user system.

3 Resiliency Metrics

Measure	Mean time to incident discovery (MTTID) [5]
Method	MTTID is the amount of time, in hours, that elapsed between the Date of Occurrence and the Date of Discovery for a given set of incidents, divided by the number of incidents
Metric	The instant value or the average value.
Use	MTTID values should trend lower over time. The value of '0 hours' indicate the hypothetical instant detection times. There is evidence the metric result may be in a range from weeks to months

Discussion

The instantaneous or the average MTTID could be grouped per types of incidents, business units, or incident severity. The NIST incident handling guide [7] recommends apply granularity to this metric by using following incident categories: DoS, Malicious code, unauthorized access, or inappropriate usage.

The implementation of this metric require to access historical data on incident related to name servers and that impact in some way the performance, availability or security of the naming service, at least at local level. Being aware that such kind of information could be subject to non-disclosure, the MTTID metric could be computed only by the owner/manager of a naming service.

Measure	Operational mean time between failures (OMTBF)
Method	OMTBF is defined as the mean value of the length of time between consecutive failures, computed as the ratio of the cumulative observed time to the number of failures under stated conditions, for a stated period of time in the life of an item. OMTBF is calculated as the sum of the operational periods divided by the number of observed failures (the operational period is defined as the difference in time between the moment the service starts operating at the normal service level until the moment the service fails). Note that the duration of the failure has no impact on the metric value.

Metric Absolute value in hours versus the target value

Use Target values depend highly on the criticality of the service and the topology of the system. Higher the value more resilient is the service (see example below)

Discussion

OMTBF should be monitored on real-time basis.

Example: If a service is very critical, the OMTBF targets will be higher compared to a normal service. As an example, the OMTBF target for an Internet service for large corporations will be higher than the target for Internet service for residential customers.

As for MTTID, also OMBTF is based on sensible information. Moreover, this metrics must be computed for failures that could really or potentially impact the naming service provision, at least at local level.

Measure Operational Availability (OA)

Method Operational availability is calculated as the percentage of the mean time that an ICT system is running at the normal service level over the total time. This metric definition is based on the definitions from [6].

Metric Percentage.

Being $MTBMS$ the Mean Time Between Maintenance Actions and MDT the Mean Down Time, $OA=MTBMS/(MTBMS+MDT)$

Use Target values for operational availability are impossible to specify for a generic ICT system. They are specified in the service level specification of the service provider.

Discussion

Operational availability is measured in a predefined time window.

For example: 99,9% operational availability measured on a yearly basis allows for a consecutive unavailability of 8,76 hours whereas the same operational availability in a measurement window of 1 month would only allow for 0,744 hours of consecutive service unavailability.

The implementation of this metric requires properly defining the concept of availability. It is important to understand that here we are not talking about the availability of a single system component, e.g. a NIC or a single server in a data center, but of a service or process, e.g. the availability of the resolution process or the availability of a transfer zone, etc...

Measure Operational reliability (OR)

Method The operational reliability of a system is a function of the Operational Mean Time between Failures (*OMTBF*) and a mission time *t*. Mission time is defined as the time between the time where the ICT system starts operating at the normal service level and the time at which the ICT system fails. Failure is defined as functioning below the acceptable service level.

The expected reliability R(t) is modeled with the exponential distribution, which describes random failures:

R(t)=e^(-t/OMTBF).

Metric Probability

Use The probability *R(t)* indicates the probability that an ICT system will run for a specified mission time *t*.

Discussion

The operational reliability of an ICT system is the ability of to perform its required functions under stated conditions (i.e. operate at the normal service level) for a specified period of time.

Target values depend highly on the criticality of the service and the topology of the ICT system. However, as soon as the metric is below *e^(-1)* (= 0,3678 = 1/e), the ICT system or service has been running longer than the mean time between failure: This means, on average, the service would have encountered a failure and failure has become more imminent.

The implementation of this metric require to compute the OMTBF and therefore are valid the same considerations.

Measure Fault report rate (*FRR*)

Method To calculate the fault report rate metric, the number of faults *NF* in a given time period *W* are counted.

FRR=(NF / W)

The time window *W* is expressed as an absolute unit of time (e.g.

hours or days) while the number of faults *N* is an absolute number, indicating how many faults have occurred in the past time window.

Metric Faults per time period.

Use No specific target can be set, as the metric value will depend on the categories of the faults that are taken into account in this metric.

Discussion

In combination with other metrics, such as operational availability, FRR can indicate the degree to which the ICT system can overcome occurring faults and maintain the normal service level.

Faults can be grouped per category or organisational departments for example.

As for the operation availability, this metric could be valuable only if fault are understood as fault of services and process (systemic point of view) and not at system level, e.g. fault of a system component such as a NIC or disk or server node.

To implement this metric must be clearly defined the set of fault that could really effect the health level of the naming system, at least at local level.

Measure Incident Rate (IR) [5]

Method The incident rate metric measures the number of security incidents that occur in a given time period from selected incident categories.

To calculate the incident rate metric, the number of security incidents *N* in a given time period *W* are counted, *IR=N/W*. The time window is expressed as an absolute unit of time (e.g. hours or days) while the number of incidents is an absolute number, indicating how many incidents have occurred in the past time window.

Metric Incident per time period

Use No specific target can be set, as the metric will also depend on the categories of incidents that are taken into account in this measure. Incident rate values should trend lower over time – assuming perfect detection capabilities. The value of 0 indicates hypothetical perfect security since there were no security incidents.

Discussion

The incident rate indicates the number of detected security incidents the organization has experienced during the metric time period. In combination with other metrics, this can indicate the level of threats, the effectiveness of security controls and/or incident detection capabilities.

Incidents can be grouped per category or organisational departments for example.

In a network of ICT security systems, it is possible that each security device reports an attack at the very same time, although only one attack is ongoing (for example: an incident on the outer firewall and an incident on the IDS system can indicate the very same event). This can result in a skewed view of the amount of incidents that occurs on the network.

The implementation of this metric requires collecting data on incidents. The evaluation of IR requires the access to sensible data.

Measure	Average Recovery speed
Method	The recovery speed metric measures the time interval between the occurring of incidents belonging to selected incident categories and the time of normal operations restore.
Metric	Mean time period between incident occurrence and service restore to normal operations
Use	No specific target can be set, as the metric will also depend on the categories of incidents that are taken into account in this measure. Average recovery speed values should trend lower over time – assuming perfect response, recovery and business continuity planning capabilities. The value of 0 indicates hypothetical perfect security since there were no security incidents.

Discussion

This metric gives an idea of how a name server or naming service is capable to react to incidents or failures and its capability to recover. The recovery speed depends on different factors ranging from technical capability of personal to adequate recovery plans but it is also influenced by prevention and preparedness policies.

Also the evaluation of this metric require to collect and access to historical and sensible data.

4 Security Metrics

As there are no standards proposing specific technical security metrics for DNS systems, the following matters were taken into consideration during the definition of this embryonic set of metrics:

- Metrics must yield quantifiable information (percentages, averages, and numbers)
- Data supporting metrics need to be readily obtainable
- Only repeatable processes should be considered for measurement
- Metrics must be useful for tracking performance and directing resources.

Security metrics can be categorized various ways. The general categories for metrics are:

- Strategic support: when the evaluation or assessment of security properties can be used to aid different kinds of decision making, such as program planning, resource allocation, and product and service selection. This is not the case of our case studies. In any case, some of the metrics that we will propose later in this report can be sued for this purpose.
- Quality assurance: when security metrics are used during the ICT development lifecycle to eliminate vulnerabilities, particularly during system design and implementation. This can include functions such as measuring adherence to secure engineering standards, identifying likely vulnerabilities that may exist, and tracking and analyzing security flaws that are eventually discovered. Again, this is not the case of our study, when we are not developing a system but only hypothesizing one based on real implementations. Also here, the metrics proposed in this report could be applied as the basis for Quality Assurance metrics during the development of future systems.
- Tactical oversight: with the aim of monitoring and reporting regarding the security status or posture of an ICT system; this can be carried out to determine compliance with security requirements (e.g., policy, procedures, and regulations), gauge the effectiveness of security controls and manage risk, provide a basis for trend analysis, and identify specific areas for improvement. This is our case, although with the caveat that we are not dealing with a real engineering architecture.

In what follows we propose a set of new metrics that, altogether, can contribute to evaluate the security readiness with respect to a possible set of attack scenarios. It's important to stress one more time that the listed metrics represent an initial effort, more intended to trigger a discussion than to be considered a definitive set.

1. **Attack Surface (ATS):**

$$ATS = \frac{|\text{Vnodes}|}{|\text{Tot\_nodes}|}$$

This metric defines the percentage of nodes of a target system/subsystem that is susceptibility to a certain type of attack. It is a primitive metric that provides a first picture of the "size of the problem". It is related to each type of attack taken into consideration.

2. **Balanced Attack Surface (BATS):**

Not all the vulnerable nodes of a system are equally "reachable" by each type of attack. For example, if an attack, to take place, needs a network connection but the target node is not connected, the impact will be minimal. This minimal impact on the total attack surface provides valuable information.

The BATS is then computed in the following way:

$$BATS = \frac{\sum_{n=1}^{n=k} R_k * |V\_nodes_k|}{|Tot\_nodes|}$$

where *K* indexes the clusters of vulnerable nodes having the same "Reachability" index (R_k). R_k is defined in the 0-1 real space.

3. **Attack Deepness (ATD)**:

$$ATD = \frac{|\mathrm{Impacted\_nodes}|}{|\mathrm{Tot\_nodes}|} .$$

Once a node is successfully attacked, the effects of the attack are usually back-propagated to other nodes. By calculating the percentage of nodes that might be indirectly affected by a successful attack on a certain node, this metric indicates how deeply the attack can affect the entire system.

4. **Balanced Attack Deepness (BATD)**:

Not all the impacted nodes have the same relevance for the "functioning capability" of the system. For that reason a more fine tuned version of the ATD index is needed to capture this aspect. As in the case of the BATS, BATD is calculated in the following way:

$$BATD = \frac{\sum_{n=1}^{n=k} Rel_k * |Impacted\_nodes_k|}{|Tot\_nodes|}$$

where *K* indexes the clusters of impacted nodes having the same "Relevance" index (Rel_k). Rel_k is defined in the 0-1 real space.

5. **System Immunity Level (SIL)**:

SIL=1-ATS

It provides in a measure of the level of protection of the system against a target attack.

6. **Attack Escalation Speed (AES)**:

The faster a successful attack is in driving the target system into the worst critical state (e.g. inability to produce the intended services), the more difficult it would be to timely react and stop the escalation trend. This metric provides a first measure of the "speed of the attack".

$$AES = \frac{Impacted\_nodes}{\Delta T}$$

Where ΔT is the time elapsed between the beginning of the attack and the reaching of the worst critical state due to the effects of the attack.

7. **Core Nodes Attack Escalation Speed (CNATES)**:

Since not all the nodes have the same level of relevance, it is possible to have a measure of the speed of the attack when damaging the core nodes of the system:

$$CNATES = \frac{Core\_Nodes\_impacted}{\Delta T}$$

8. **Node Unplanned Downtime Impact (NUDI)**:

This metric, measured in monetary currencies (euro, dollar, etc.), quantifies the economical damage of the downtime of an attacked node.

9. **Total Unplanned Downtime Impact (TUDI)**:

$$TUDI = \sum_{i=1}^{i=n} NUDI_i$$

where *i* indexes the nodes which went down as a result of the attack. It adds up all NUDI metrics for the relevant nodes.

10. **Support Response Time (SRT):**

It is the mean time taken by the security team to detect the attack and start a mitigation procedure, measures from the first symptoms of the attack are identified until the attacking process is set under control.

11. **Node Mean Time to Recovery (NMTR):**

It is the mean time needed to restore the attacked node to an acceptable situation. It is measured since the node loses the capability to deliver its intended services, until the node is back into an acceptable state.

12. **System Mean time to Recovery (SMTR):**

It is the mean time needed to completely sanitize the entire system after a successful attack. It is measured since the system loses the capability to deliver its intended services, until it is put back in operative state.

13. **Vulnerability Density (VD):**

It characterizes the incidence rate of security breaches in the nodes.

$$VD = \frac{n\_vuln}{Tot\_nodes}$$

14. **Weighted Vulnerability Density (WVD):**

Weights can be inserted in the VD formula to capture the relevance of the vulnerabilities. This relevance can be measured in light of the intended services of the system, taken into account one by one or all together.

15. **Annualized Loss Expectancy (ALE):**

The ALE is the monetary loss that can be expected for an asset due to the risk of its loss during one-year period. It is defined as

*ALE= SLE * ARO*

where *SLE* is the Single Loss Expectancy (which can be quantified as the TUDI + Recovery Cost), while *ARO* is the annualized rate of occurrence, i.e. the number of successful exploitation that a target attack can accomplish in one year without being countered by specific protection measures.

16. **Business Adjusted Risk (BAR)**:

BAR is a technique proposed by Gary McGraw to classify security defects by their vulnerability type, degree of risk and potential business impact. When assessing a system, a subsystem or a node, for each security defect, a BAR score is calculated as follows:

BAR=Business_impact ×risk_of_exploit;

where the business_impact and the risk_of_exploit are defined in the integer space (1-5). The first term of the form indicates the damage that would be sustained if the defect were exploited (5 represent a flaw which could cause a significant financial impact). The second term of the formula, indicates how easily an attacker can exploit the given defect (5 denotes high-risk).

5 Bibliography

[1] Lihua Yuan, Kant, K., Mohapatra, P., Chen-Nee Chuah; UC, Davis, A Proxy View of Quality of Domain Name Service, INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE, 6-12 May 2007

[2] A. Huber, R. van Mook, Measures for making DNS more resilient against forged answers, Network WG IETF, RFC 5452, Jan 2009

[3] ICANN, Security, Stability and Resiliency of the Domain Name System, Jan. 2009 http://www.gtisc.gatech.edu/pdf/DNSSSRPaper.pdf

[4] Mark Santcroos, Olaf M. Kolkman, DNS Threat Analysis, NLnet Labs document 2006-SE-01 version 1.0 May 3, 2007

[5] 'The CIS security metrics - Consensus Metric Definitions v1.0.0', The Center of Internet Security, 2009;

[6] 'Complete Guide to Security and Privacy Metrics: Measuring Regulatory Compliance, Operational Resilience, and ROI', Debra S. Hermann, 2007;

[7] The NIST incident handling guide (NIST Special Publication 800-61 Revision 1) http://csrc.nist.gov/publications/nistpubs/800-61-rev1/SP800-61rev1.pdf;

[8] IEEE 90 – Institute of Electrical and Electronics Engineers, IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries. New York, NY: 1990

[9] Secure Domain Name System (DNS) Deployment Guide, NIST, April 2010, http://csrc.nist.gov/publications/nistpubs/800-81r1/sp-800-81r1.pdf

[10] Eric Osterweil, Dan Massey, Lixia Zhang, The Design of Metrics for Quantifying the DNSSEC Deployment

[11] R. Villamarín-Salomón, J. Brustoloni, Identifying Botnets Using Anomaly Detection Techniques Applied to DNS Traffic

[12] Measuring the health of the Domain Name System, Report of the 2nd Annual Global Symposium on DNS Security, Stability and Resiliency, http://www.icann.org/en/topics/ssr/dns-ssr-symposium-report-1-3feb10-en.pdf

[13] J.Damas, F. Neves. RFC 5358/BCP 140, Preventing Use of Recursive Nameservers in Reflector Attacks