基于分散化序列的联网 ICS 设备搜索技术

刘知竹[1],冯 璐[2],荀 鹏[1],刘吉元[1]
(1. 国防科技大学 计算机学院,湖南 长沙 410073;
2. 长沙学院 电子信息与电气工程学院,湖南 长沙 410022)

摘 要:针对已知的联网设备搜索行为存在重复扫描的问题,提出了一种基于分散化序列的联网 ICS 设备搜索技术,并设计了分散化序列的启发式生成算法。对于分散化序列,邻近的 IPv4 地址在其中的位置相距较远,使得按照其顺序执行扫描时能够降低对小规模网段的扫描频度。实验测试了基于 Modbus 和 Siemens S7 协议获取 ICS 设备信息的机制,分析了生成分散化序列的最优算法参数,模拟实现了对 IPv4 空间中的 ICS 设备搜索并从蜜罐视角分析了搜索行为的特征。实验结果表明,基于该搜索技术能够在分布式扫描全网 ICS 设备的同时避免重复扫描,提高了搜索的效率。

关键词:工业控制系统;联网设备搜索;分散化序列;Modbus;Siemens S7
中图分类号:TP393 **文献标识码**:A **文章编号**:1673-629X(2018)11-0001-05
doi:10.3969/j.issn.1673-629X.2018.11.001

Networked ICS Device Search Technique Based on Dispersed Sequence

LIU Zhi-zhu[1],FENG Lu[2],XUN Peng[1],LIU Ji-yuan[1]
(1. School of Computer,National University of Defense Technology,Changsha 410073,China;
2. School of Electronic Information and Electrical Engineering,Changsha University,Changsha 410022,China)

Abstract:Aiming at the problem of repeated scanning in searching behavior of known networked devices,we present a networked ICS device search technology based on dispersed sequences,and design a heuristic generation algorithm for dispersed sequences. Nearby IPv4 addresses are far apart in the dispersed sequence,which decreases the frequency of scanning in the order of the sequence on small scale network. The mechanism of obtaining ICS device information based on Modbus and Siemens S7 protocol is tested in the experiment,the optimal algorithm parameters for generating dispersed sequences are analyzed,the ICS device search in IPv4 space is simulated,and the characteristics of search behavior are analyzed from the perspective of honeypot. The experiment shows that based on this search technology,the whole network ICS device can be scanned in distributed while avoiding repeated scanning.

Key words:industrial control system;networked device search;dispersed sequences;Modbus;Siemens S7

0 引 言

作为关键基础设施的工业控制系统(industrial control system,ICS)在网络空间中是重要的资源,其包含监控和数据采集系统(supervisory control and data acquisition,SCADA)、可编程逻辑控制器(programmable logic controller,PLC)等,在电气、化工以及核能等领域应用广泛[1]。与传统的 IT 系统相比,工业控制系统具有信息物理融合性质[2],主要面向对生产过程的控制而非对信息的管理,因此 ICS 的安全与生产生活关系更为密切。近年来,通过全网扫描的方式,大量的在线应用及物理设备被发现,其中也包括 ICS 设备[3-4]。对于被探测发现的 ICS 设备,一方面可以基于漏洞扫描工具等分析该类设备的网络安全态势[5];另一方面,由于部分 ICS 通信协议的认证机制不完善,外网节点基于相应协议下的特定指令也可获得设备信息,因此可以提取该类设备的属性并在此基础上展开数据统计与分析,这是网络空间测绘的一种有效方法[6-8]。

全网扫描的一种典型应用是联网设备搜索引擎,目前有国外的 Censys、Shodan,以及国内的 ZoomEye、O'Shadan 和 FOFA 等,它们将网络中的应用与设备信息汇集形成数据库,为相关研究提供参考。Censys 来自密歇根大学的 Internet-Wide Scanning Research 项目,其使用的 ZMap 扫描器以及 ZDb 数据库性能高于

收稿日期:2017-11-23 **修回日期**:2018-03-07 **网络出版时间**:2018-06-29
基金项目:国家自然科学基金(61572514)
作者简介:刘知竹(1995-),男,硕士研究生,研究方向为信息物理系统安全;冯 璐,通信作者,副教授,研究方向为通信网络技术。

同类工具,具有较高的工作效率[9-10]。Shodan 是最早出现的联网设备搜索引擎,其扫描节点跨地域分布,并且对 ICS 设备具有成熟的搜索能力[11-12]。这些搜索引擎均能发现 ICS 设备,因此可以基于工控蜜罐分析它们的行为特征。根据文献[12]的统计,大部分搜索行为存在重复扫描的情况,即单个扫描节点高频度地扫描目标,或者多个扫描节点在短期内扫描同一个目标;此外,针对不同 ICS 协议的扫描频率也不同。这都使得搜索效率降低,同时搜索行为也容易被发现。

基于此,文中提出一种基于分散化序列的联网 ICS 设备搜索技术,能够在搜索整个 IPv4 网络空间中 ICS 物理设备的同时,避免短期内对单个 IPv4 地址及其 TCP 端口的重复扫描。同时,扫描节点在按照经过分散化的 IPv4 地址顺序依次扫描时,能够在短时间内对一个小规模网段仅扫描其中一个 IPv4 地址。文中对分散化序列进行了参数定义,并在此基础上设计了一种分散化序列的启发式生成算法,并通过实验对其进行分析与验证。

1　联网 ICS 设备搜索机制

由于 ICS 通信协议位于特定的 TCP 端口上,因此通过端口扫描可以初步判定目标是否属于 ICS 设备,然后基于对应协议获取目标信息。为了搜索整个 IPv4 空间中所有 ICS 设备,需要扫描每个地址下的所有 ICS 协议端口。假设全网扫描由多个分布式扫描节点完成,则可以将 IPv4 地址分片后均衡地调度给它们,扫描任务的集合可以表示为矩阵[$\boldsymbol{w}_{ij}$],如图 1 所示。

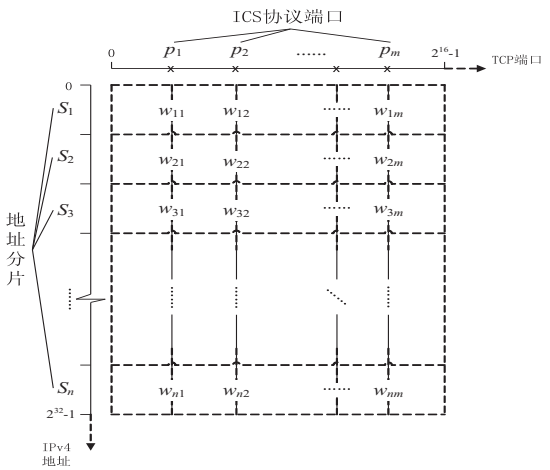


图 1　扫描任务集合

其中 IPv4 地址被均分为 $\{S_i \mid 1 \leqslant i \leqslant n\}$,分片长度为 l,各种 ICS 协议的 TCP 端口集合为 $\{p_j \mid 1 \leqslant j \leqslant m\}$,$\boldsymbol{w}_{ij}$ 是地址与端口序偶的向量,其表达式为:

$$\boldsymbol{w}_{ij} = [< \mathrm{addr}_{l(i-1)+1}, p_j > \cdots < \mathrm{addr}_{li}, p_j >]^{\mathrm{T}} \quad (1)$$

扫描节点不断地选取序偶<addr, p>执行扫描,可以等价为从它负责的分片 S 的一个置换上依次取出地址 addr,然后扫描其端口 p。若这一过程能够确保不在短时间内扫描邻近的多个地址,即邻近的地址在置换后的地址序列上相距较远,则称这段地址序列具备分散化性质。分散化序列的启发式生成算法在第 2 节阐述,该算法对连续地址序列 S 进行处理后能生成满足分散化性质的序列 $Q(S)$ 和剩余序列 $R(S)$。对于每个任务 w_{ij},经算法处理后余下的剩余地址与端口序偶构成剩余任务 μ_{ij},并定义补充任务 δ_j 保存剩余任务矩阵[$\boldsymbol{\mu}_{ij}$]的列向量,其表达式为:

$$\boldsymbol{\mu}_{ij} = [< \mathrm{addr}_{n_1}, p_j > \cdots < \mathrm{addr}_{n_r}, p_j >]^{\mathrm{T}}, \quad l(i-1)+1 \leqslant n_1 < \cdots < n_r \leqslant li \quad (2)$$

$$\boldsymbol{\delta}_j = [\boldsymbol{\mu}_{1j} \cdots \boldsymbol{\mu}_{nj}]^{\mathrm{T}} \quad (3)$$

此外,将[$\boldsymbol{w}_{ij}$]的同行元素随机互换后变为[$\boldsymbol{w}'_{ij}$],此时每个任务 $\boldsymbol{w}'_{ij}$ 中的地址仍属于 S_i,但端口不再相同,这样扫描节点在对分片 S_i 的扫描过程中不会只扫描一种端口。随机互换后的任务表达式为:

$$\boldsymbol{w}'_{ij} = [< \mathrm{addr}_{l(i-1)+1}, p_j^{l(i-1)+1} > \cdots < \mathrm{addr}_{li}, p_j^{li} >]^{\mathrm{T}} \quad (4)$$

$$\boldsymbol{\mu}'_{ij} = [< \mathrm{addr}_{n_1}, p_j^{n_1} > \cdots < \mathrm{addr}_{n_r}, p_j^{n_r} >]^{\mathrm{T}}, \quad l(i-1)+1 \leqslant n_1 < \cdots < n_r \leqslant li \quad (5)$$

$$\boldsymbol{\delta}'_j = [\boldsymbol{\mu}'_{1j} \cdots \boldsymbol{\mu}'_{nj}]^{\mathrm{T}} \quad (6)$$

对于任务集合[$\boldsymbol{w}'_{ij}$],基于负载均衡的原则将任务 $\boldsymbol{w}'_{ij}$ 按列主序随机调度给扫描节点。当 $\boldsymbol{w}'_{ij}$ 调度给扫描节点后,首先基于算法生成普通任务 $Q'(\boldsymbol{w}'_{ij})$,然后从中依次取出地址扫描相应端口,同时将剩余任务 $R'(w'_{ij}) = \boldsymbol{\mu}'_{ij}$ 加入到补充任务 $\boldsymbol{\delta}'_j$ 中。当补充任务 $\boldsymbol{\delta}'_j$ 与普通任务的工作量相同或者已调度完一列的任务时,将 $\boldsymbol{\delta}'_j$ 中存放的剩余任务 $\{\boldsymbol{\mu}'_{kj}\}$ 调度给扫描节点,然后清空 $\boldsymbol{\delta}'_j$。扫描节点不断遍历 $\{\boldsymbol{\mu}'_{kj}\}$ 中的任务 $\boldsymbol{\mu}'_{kj}$,并每次从中随机取出序偶<addr, p>执行扫描。

对于任务调度机制,设扫描节点集为 A,每个节点 α 的实时并行任务数量为 $\mathrm{thd}(\alpha)$,上限为 $\mathrm{mthd}(\alpha)$,则每次调度任务时,从可执行任务并且 thd 最小的扫描节点集合 A' 中随机选取节点分配任务。该节点集合的表达式为:

$$A' = \{\alpha \mid \mathrm{thd}(\alpha) = \min_{\beta \in A} \mathrm{thd}(\beta) < \mathrm{mthd}(\alpha)\} \quad (7)$$

综上,全网扫描的过程如下:

步骤 1:将[$\boldsymbol{w}_{ij}$]的同行元素的顺序随机化为[$\boldsymbol{w}'_{ij}$],设置 $i \leftarrow 1$, $j \leftarrow 1$,初始化 δ_k 为空,其中的 $1 \leqslant k \leqslant m$。

步骤 2:对 w'_{ij} 生成序列 $Q'(w'_{ij})$ 和 $R'(w'_{ij})$,并将 $R'(w'_{ij})$ 作为 μ'_{ij} 加入 δ'_j,调用扫描节点对 $Q'(w'_{ij})$ 进行扫描。

步骤 3:若 i 等于 n,则调用扫描节点对 $\delta_j^{'}$ 扫描后清空 $\delta_j^{'}$,设置 $i \leftarrow 1$, $j \leftarrow j+1$,若 j 等于 $m+1$ 则返回步骤 1;若 i 不等于 n,则设置 $i \leftarrow i+1$,若 $\delta_j^{'}$ 的地址数大于等于 1,则调用扫描节点对 $\delta_j^{'}$ 进行扫描后清空 $\delta_j^{'}$。

步骤 4:返回步骤 2。

2　分散化序列及其生成算法

首先对分散化序列进行定义。设 S 为一段具有单调性的序列,若 $Q(S)$ 为其置换并满足:(1) $Q(S)$ 中的相邻元素在 S 中的数值差距大于 s_1;(2) S 中任何数值差距小于 s_2 的元素在 $Q(S)$ 中的位置相距大于 s_3。则称 $Q(S)$ 为 S 的一个满足参数$<s_1, s_2, s_3>$的分散化序列。

基于该定义,在设置一定参数的条件下,当扫描节点从 $Q(S)$ 依次取地址扫描时,能够在遍历 S 中所有地址的同时降低对小规模网段的扫描频度。其中 s_1 和 s_2 表征了扫描行为在空间上的分散化程度,s_3 表征了扫描行为在时间上的分散化程度。参数值越高则分散化程度越好,但是参数值过高会使得相应的置换序列不存在。

这里给出一种对分散化地址序列的启发式生成算法,能够快速地产生满足参数条件的序列 $Q(S)$ 以及剩余地址序列 $R(S)$。当选取合适的参数时,能够使得 $R(S)$ 的元素数量较少,从而能够使扫描行为满足分散化条件的同时近似地覆盖对 S 的扫描。假设扫描节点扫描一次用时为 Δt,对小规模网段的扫描时间间隔为 t,则算法的基本流程为:

输入:地址序列 S,序列长度 l,距离间隔 s_1、s_2,单次扫描用时 Δt,时间间隔 t。

步骤 1:初始化地址序列 $Q \leftarrow [\,]$,序列 $G_1 \leftarrow [\,]$,序列 $G_2 \leftarrow [\,]$,距离 $s_3 \leftarrow \lfloor t/\Delta t \rfloor$,地址 $q \leftarrow \theta$。

步骤 2:将 G_2 中的所有元素减 1。

步骤 3:若 G_2 中存在元素为 0,则将 G_1 和 G_2 中相应下标的两个元素同时去除。

步骤 4:若 $q \neq \theta$,取邻域 $N_{s_1}(q) = (q - s_1, q + s_1)$,若 l 为 0 或 $S - N_{s_1}(q)$ 为空,输出 Q 以及 $R \leftarrow S$,退出。

步骤 5:取集合 $P = S - N_{s_1}(q) - \cup N_{s_2}^{'}(g_i)$,其中 $g_i \in G_1$ 且 $N_{s_2}^{'}(g_i) = (g_i - \lfloor s_2/2 \rfloor, g_i + \lfloor s_2/2 \rfloor)$。若 P 为空,则设置 $q \leftarrow \theta$,并向 Q 中加入 θ;若 P 非空,则从 P 中随机选取地址 addr 加入 Q,并从 S 中去除 addr,分别将 addr 和 s_3 加入到 G_1 和 G_2,设置 $l \leftarrow l-1$。

步骤 6:返回步骤 2。

输出:地址序列 Q,剩余地址序列 R。

扫描节点在遍历 $Q(S)$ 时,若取出了有效地址 addr,则对其端口 p 执行扫描;若取出了无效地址 θ,则待机 Δt。

3　实验分析

本节对提出的联网 ICS 设备搜索技术进行实验分析。首先测试相关 ICS 协议下获取信息的机制,然后分析分散化序列生成算法的参数对序列效果的影响,最后模拟全网扫描并统计扫描行为的特征。

3.1　协议交互测试

该小节测试 Modbus 和 Siemens S7 两种协议下获取 ICS 设备信息的机制,以验证在发现 ICS 协议端口开启后能够提取出设备信息。

3.1.1　Modbus

Modbus 协议工作在 TCP 的 502 端口,其报文类型分为请求、应答和异常应答,报文内容由地址码、功能码和数据组成[13]。其中地址码标识 Modbus 网络中的设备单元,功能码说明指令能够实现的功能,包括获取设备信息、读写设备的存储器等。根据文献[13],基于 0x2B 功能码并设置子功能码为 0x0E 时可请求设备信息,基于 0x11 功能码可读取 Modbus 网络中设备单元的名称和运行状态。协议交互的基本流程为:

步骤 1:依次向地址码为 0x01 至 0x20 的对应单元发送功能码为 0x2B,子功能码为 0x0E,描述类别参数为 0x01 的报文,请求目标回复基本的设备信息。若某个地址码对应的单元回复了设备信息,则向该单元再分别发送描述类别参数为 0x02 和 0x03 的报文,尝试获取更详细的设备信息,然后结束该步骤。

步骤 2:依次向地址码为 0x01 至 0xFF 的对应单元发送功能码为 0x11 的请求报文,请求目标回复名称和运行状态。

从搜索引擎 FOFA 选取 150 个已知开启 502 端口的 IPv4 地址,基于上述机制进行交互。结果显示 141 个地址回复了设备信息,样例如表 1 所示。

表 1　某 IPv4 地址回复的 Modbus 设备部分信息

属性	设备信息
Major Minor Revision	05.20
Product Code	TWDLCAE40DRF
Product Name	TWID0
User Application Name	Mon Twido
Vendor Name	TELEMECANIQUE

3.1.2　Siemens S7

Siemens S7 协议工作在 TCP 的 102 端口,并位于 TPKT 和 COTP 协议之上[14]。当设置功能码为 0x43 并且子功能码为 0x01 时,可以请求设备返回各类程序块(Block)的数量,这些程序块中存放着设备的代码与数据。当设置功能码为 0x44 并且子功能码为 0x01 时,可以读取系统状态列表(system status list, SSL)[15],其中描述类别参数为 0x0011 时能够获取保存模块标识(module identification)的 SSL;描述类别参

数为 0x001c 时能够获取保存组件标识(component identification)的 SSL,其中均包括设备信息。协议交互的基本流程为:

步骤 1:发送功能码为 0x44,子功能码为 0x01,描述类别参数为 0x0011 的报文,请求目标回复关于模块标识的系统状态列表。

步骤 2:发送功能码为 0x44,子功能码为 0x01,描述类别参数为 0x001c 的报文,请求目标回复关于组建标识的系统状态列表。

步骤 3:发送功能码为 0x43,子功能码为 0x01 的报文,请求目标回复程序块数量的统计信息。

从搜索引擎 FOFA 选取 150 个已知开启 102 端口的 IPv4 地址,基于上述机制进行交互。结果显示 140 个地址回复了设备信息,样例如表 2 所示。

表 2　某 IPv4 地址回复的 Siemens S7 设备部分信息

属性	设备信息
Copyright	Original Siemens Equipment
Plant Identification	Mouser Factory
Module Type Name	IM151-8 PN/DP CPU
Name of the PLC	Technodrome
Name of the Module	Siemens, SIMATIC, S7-200

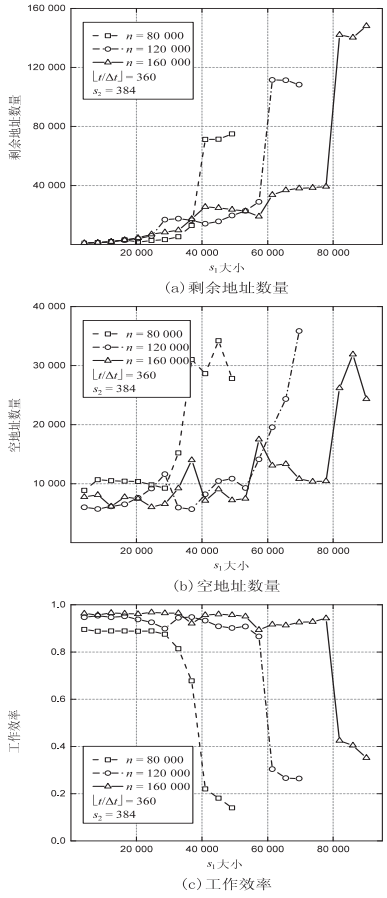
3.2　算法参数分析

该小节分析参数对分散化序列生成效果的影响以及最优参数的取值,显然参数的值越高,分散化的效果越好,但是过高会影响算法,使得序列中的空地址数量 $|\Theta|$ 以及剩余地址数量 $|R|$ 过多。为了评测序列的生成效果,考虑在执行普通任务 $Q'(w)$ 时扫描有效地址用时占总任务用时的百分比 r,其反映了执行任务时的工作效率。

$$r = 1 - \frac{|\Theta| \cdot \Delta t}{(n - |R| + |\Theta|) \cdot \Delta t} = \frac{n - |R|}{n - |R| + |\Theta|} \tag{8}$$

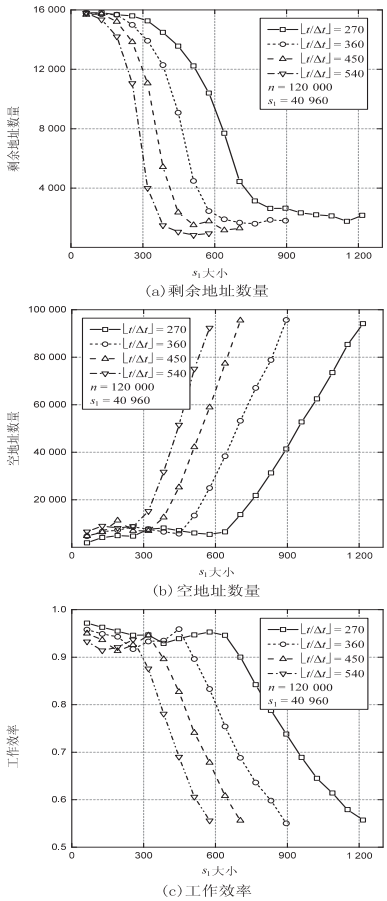
首先分析参数 s_1,选取 3 组初始参数后调节 s_1 观察序列效果的变化,每组参数下运行 5 次后取平均值,结果如图 2 所示。当 s_1 位于序列地址总数 n 的一定比例之前 $|R|$ 逐渐增长,$|\Theta|$ 在一定区间内变动;一定比例之后 $|R|$ 和 $|\Theta|$ 急剧提高,此时 r 也迅速降低。这说明在不超过一定阈值的情况下增大 s_1 对序列效果不会造成明显的影响,并且阈值位于 n ·40% 左右。

接下来分析参数 s_2 和 $s_3 = \lfloor t/\Delta t \rfloor$,选取 4 组初始参数后调节 s_2 观察序列效果的变化,每组参数下运行 5 次后取平均值,结果如图 3 所示。随着 s_2 的增长,$|\Theta|$ 和 r 起初保持稳定,在 s_2 超过一定阈值之后开始线性变化,并且 s_3 越大阈值越小,同时变化速度越快;$|R|$ 非线性递减,超过一定阈值后再次保持稳定,并且 s_3 越大阈值越小,同时递减速度越快。因此,可以发现 s_2 和 s_3 位于阈值点时算法效果最好。



(a) 剩余地址数量　(b) 空地址数量　(c) 工作效率

图 2　相关指标随 s_1 增长时的变化



(a) 剩余地址数量　(b) 空地址数量　(c) 工作效率

图 3　相关指标随 s_2 增长时的变化

3.3　模拟搜索测试

该小节基于文中的搜索技术初步模拟了全网扫描,并统计对蜜罐地址的扫描行为。实验不考虑私有 IPv4 地址等特殊情况,认为所有 IPv4 地址均具备扫描价值。假设使用 10 个扫描节点在 IPv4 空间中搜索 15 种 ICS 协议设备,其中每种协议位于不同 TCP 端口上;每个扫描节点拥有 1 200 个扫描线程,执行一次端口扫描用时 $\Delta t = 0.5$ s;地址分片长度 $l = 131\ 072$,参数 $s_1 = 81\ 920$, $s_2 = 640$, $s_3 = \lfloor t/\Delta t \rfloor = 270$。

首先随机生成一个地址 addr,然后基于算法生成 45 次序列,统计每次产生的剩余地址数和空地址数,以及 addr 在生成序列中的位置。根据剩余地址数和空地址的平均情况,模拟出整个扫描任务矩阵以及补充任务集合,然后将任务依次调度给扫描节点,每次调度时从空闲线程数最多的扫描节点中随机选取。最后模拟了 3 轮全网扫描,并从中取出 addr 所在的 45 个任务,统计出它被扫描时的时间点以及扫描节点,结果如图 4 所示。

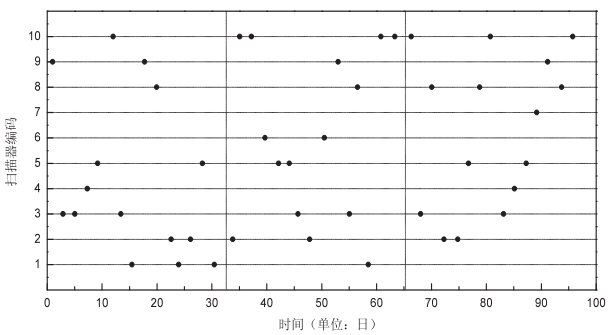


图 4　模拟扫描活动统计

在每轮全网扫描中,addr 的 15 个 ICS 端口分别被扫描 1 次,并且扫描的时间间隔均衡,搜索效率较高。全网扫描的周期可以通过参数调整,在此次模拟中为 32.6 天。

4　结束语

针对已有联网设备搜索对 ICS 设备重复扫描的问题,提出一种基于分散化序列的联网 ICS 设备搜索技术,设计出分散化序列的启发式生成算法,并通过实验测试了两种 ICS 协议下的设备信息获取机制,分析了算法参数对生成的序列效果影响,模拟了全网扫描并统计出对单个 IPv4 地址的扫描行为特点。实验结果表明,该方法能够在分布式扫描全网 ICS 设备的同时避免重复扫描,提高了搜索效率。

参考文献:

[1] NIST SP800-82. Guide to industrial control systems (ICS) security[S]. USA:National Institute of Standards and Technology,2011.

[2] 彭　勇,江常青,谢　丰,等. 工业控制系统信息安全研究进展[J]. 清华大学学报:自然科学版,2012,52(10):1396-1408.

[3] DURUMERIC Z,BAILEY M,HALDERMAN J A. An Internet-wide view of internet-wide scanning[C]//Proceedings of the 23rd USENIX conference on security symposium. San Diego,CA:USENIX Association,2014:65-78.

[4] VAVRA J,HROMADA M. Possibilities of the search engine shodan in relation to SCADA[C]//International conference on emerging security information,systems and technologies. [s.l.]:[s.n.],2016:130-135.

[5] EL M,MCMAHON E,SAMTANI S,et al. Benchmarking vulnerability scanners:an experiment on SCADA devices and scientific instruments[C]//IEEE international conference on intelligence and security informatics. Beijing,China:IEEE,2017:83-88.

[6] 王欢欢,张冬梅,于　亮. 一种针对工控系统的网络探测方法[C]//全国青年通信学术年会. 北京:国防工业出版社,2014:19-23.

[7] LEVERETT E P. Quantitatively assessing and visualising industrial system attack surfaces[D]. UK:University of Cambridge,2011.

[8] 赵　帆,罗向阳,刘粉林. 网络空间测绘技术研究[J]. 网络与信息安全学报,2016,2(9):1-11.

[9] DURUMERIC Z,ADRIAN D,MIRIAN A,et al. A search engine backed by internet-wide scanning[C]//Proceedings of the 22nd ACM SIGSAC conference on computer and communications security. Denver,Colorado,USA:ACM,2015:542-553.

[10] DURUMERIC Z,WUSTROW E,HALDERMAN J A. ZMap:fast internet-wide scanning and its security applications[C]//Proceedings of the 22nd USENIX conference on security. Washington,D. C.:USENIX Association,2013:605-620.

[11] BODENHEIM R,BUTTS J,DUNLAP S,et al. Evaluation of the ability of the shodan search engine to identify internet-facing industrial control devices[J]. International Journal of Critical Infrastructure Protection,2014,7(2):114-123.

[12] 灯塔实验室. 针对网络空间关键基础设施情报收集的组织行为分析报告[R]. 北京:灯塔实验室,2016.

[13] Modbus Organization. Modbus application protocol specification[S]. USA:Modbus Organization,2012.

[14] KLEINMANN A,WOOL A. Accurate modeling of the Siemens S7 SCADA protocol for intrusion detection and digital forensics[J]. Journal of Digital Forensics Security & Law,2014,9(2):37-50.

[15] Siemens. Systems software for S7-300/400 system and standard functions[S]. GER:Siemens,2007.