基于时序逻辑的3种网络攻击建模

聂 凯 周清雷 朱维军 张朝阳
(郑州大学信息工程学院 郑州 450001)

摘 要 与其他检测方法相比,基于时序逻辑的入侵检测方法可以有效地检测许多复杂的网络攻击。然而,由于缺少网络攻击的时序逻辑公式,该方法不能检测出常见的back,ProcessTable以及Saint 3种攻击。因此,使用命题区间时序逻辑(ITL)和实时攻击签名逻辑(RASL)分别对这3种攻击建立时序逻辑公式。首先,分析这3种攻击的攻击原理;然后,将攻击的关键步骤分解为原子动作,并定义了原子命题;最后,根据原子命题之间的逻辑关系分别建立针对这3种攻击的时序逻辑公式。根据模型检测原理,所建立的时序逻辑公式可以作为模型检测器(即入侵检测器)的一个输入,用自动机为日志库建模,并将其作为模型检测器的另一个输入,模型检测的结果即为入侵检测的结果,从而给出了针对这3种攻击的入侵检测方法。

关键词 命题区间时序逻辑,实时攻击签名逻辑,模型检测,入侵检测
中图法分类号 TP301 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2018.02.036

Modeling for Three Kinds of Network Attacks Based on Temporal Logic

NIE Kai ZHOU Qing-lei ZHU Wei-jun ZHANG Chao-yang
(School of Information Engineering,Zhengzhou University,Zhengzhou 450001,China)

Abstract Compared with other detection methods,the intrusion detection methods based on temporal logic can detect many complex network attacks effectively. There is no network attack temporal logic formula,so common back,ProcessTable and Saint attacks can not be detected using the above method. Thus,this paper employed propositional interval temporal logic (ITL) and real-time attack signature logic (RASL) to model the temporal logic formula for the three attacks,respectively. In general,based on attack basic principle of the three attacks,the key attack steps are decomposed into atomic actions. Next,this paper defined atomic propositions. Lastly,according to the relationship between the atomic propositions,this paper constructed the network attack temporal logic formula which is an input of the model checker. In addition,the automaton was used to model the log library as another input of the model checker. The output of the model checker is the result of intrusion detection in the three network attacks. Besides,the intrusion detection method for three attacks was given.

Keywords Propositional interval temporal logic,Real-time attack signature logic,Model checking,Intrusion detection

1 引言

入侵检测(ID)是一种重要的网络安全技术,根据检测原理的不同,被分为异常检测和误用检测。由于异常检测的误报率较高,因此部署的入侵检测系统大部分都采用误用检测。误用检测根据已知的入侵或攻击方式的行为特征,建立攻击模式库或入侵特征库,然后通过分析被检测系统中的数据,采用模式匹配算法判断被检测系统中是否存在攻击或入侵行为。误用检测可以检测出已知模式或特征的入侵行为,但很难准确地识别新的攻击模式,因为随着攻击手段的不断变化,无法在一个系统中对所有的攻击行为进行特征提取。

为此,基于模型检测的入侵检测技术应运而生。与基于模式匹配的方法相比,基于模型检测的方法可以有效地描述不断变化的攻击模式[1-4]。模式匹配通常适用于检测数据之间的不一致性,而模型检测中的自动机、时序逻辑等技术适用于检测行为的不一致性。因此,基于模型检测的方法比基于模式匹配的方法具有更强的检测能力,因为除了相对简单的数据之外,入侵攻击涉及更复杂的行为[4]。模型检测的基本原理[5]可概括如下:1)使用时序逻辑公式描述攻击模式,使用自动机记录审计日志库中发生的情况;2)使用模型检测算法检测自动机是否满足公式(即日志库中的记录是否匹配攻击模式)。时序公式中的逻辑运算符可以灵活地描述攻击动作之间的各种逻辑关系。

然而,当前的基于模型检测的入侵检测方法仍然存在若干需要解决的问题。首先,该技术不能完全检测到公认标准入侵检测数据集KDD CUP 99中常见的39种攻击[5],因为

到稿日期:2016-11-29 返修日期:2017-02-07 本文受国家重点研发计划(2016YFB0800100),国家自然科学基金(U1204608,U1304606,61572444),中国博士后科学基金(2015M572120,2012M511588)资助。

聂 凯(1987-),男,硕士生,主要研究方向为网络安全,E-mail:ieknie@163.com;周清雷(1962-),男,博士,教授,博士生导师,主要研究方向为网络安全,E-mail:ieqlzhou@zzu.edn.cn(通信作者);朱维军(1976-),男,博士,副教授,主要研究方向为网络安全;张朝阳(1983-),男,硕士,主要研究方向为网络安全。

它缺乏描述这些攻击的时序逻辑公式。

为了解决该问题，文献[6]使用线性时序逻辑为上述 39 种攻击中的 13 种攻击建立了线性时序逻辑公式；文献[7]使用时间区间时序逻辑为上述 39 种攻击中的 11 种攻击建立了时间区间时序逻辑公式；文献[8]在此二者的基础上构建了基于模型检测的入侵检测基准测试平台，该平台能够对已有时序逻辑公式的 22 种攻击实施检测。然而，对于上述 39 种攻击中的 15 种攻击，目前还没有相关工作为其建立时序逻辑公式，因此仍然无法使用基于模型检测的入侵检测方法对其进行检测。鉴于此，本文使用时序逻辑中的命题区间时序逻辑(ITL)和实时攻击签名逻辑(RASL)来对余下的 15 种攻击中的 back，ProcessTable 以及 Saint 这 3 种攻击进行时序逻辑建模，为基于模型检测的入侵检测方法奠定基础。

2 基础知识

本节介绍命题区间时序逻辑(ITL)和实时攻击签名逻辑(RASL)的定义，包括基本语法规则和语义，为下文的建模奠定理论基础。

2.1 命题区间时序逻辑(ITL)

定义 1 命题区间时序逻辑(Interval Temporal Logic，ITL)的语法规则[9-12]：

1)对于所有的 $p\in AP$，p 是一个 ITL 公式。

2)如果 φ,ψ 是 ITL 公式，那么构造的 $\neg\varphi,\varphi\vee\psi,\bigcirc\varphi,\varphi;\psi,\varphi^*$ 都是 ITL 公式。

定义 2(ITL 语义) 令 $c\in N$，$s_p(k)$ 表示 $p\in AP$ 在状态 s_k 下的真值。

ITL 满足的关系被归纳定义如下：

1)$I|=p$ 当且仅当 $s_p(k)=\text{true}$。

2)$I|=\neg\varphi$ 当且仅当 $\neg(I|=\varphi)$。

3)$I|=\varphi_1\vee\varphi_2$ 当且仅当 $I|=\varphi_1$ or $I|=\varphi_2$。

4)$I|=skip$ 当且仅当 $len(\sigma)=1$。

5)$I|=\bigcirc\varphi$ 当且仅当 $(\sigma,i,k+1,j)|=\varphi$。

6)$I|=\varphi_1;\varphi_2$ 当且仅当 $\exists r,k\leqslant r\leqslant j$，$s$ 使得 $(\sigma,i,k,r)|=\varphi_1$ 且 $(\sigma,i,k,r)|=\varphi_2$。

7)$I|=\varphi^*$ 当且仅当：①存在无限多 $r_0,\cdots,r_n\in N_\omega$，使得 $k=r_0\leqslant r_1\leqslant\cdots\leqslant r_{n-1}\leqslant r_n=j$，$(\sigma,i,k,r_0)|=\varphi$，且对于每一个 $l(1\leqslant l\leqslant n)$，$(\sigma,r_{l-1},r_{l-1},r_l)|=\varphi$；②$k=j$。

定义 3(ITL 导出公式) $\Diamond\varphi=\text{true};\varphi$，$\Box\varphi=\neg\Diamond\neg\varphi$，$\varphi\wedge\psi=\neg(\neg\varphi\vee\neg\psi)$，$\varphi_1\parallel\varphi_2=\varphi_1\wedge(\varphi_2;\text{true})\vee\varphi_2\wedge(\varphi_1;\text{true})$。

2.2 实时攻击签名逻辑(RASL)

定义 4(实时攻击签名逻辑，Real-time Attack Signature Logic，RASL)[12-14] 公式具有在 Backus-Naur 表中给出的以下语法：

1)Terms $t::=T|T_f$。

2)约束公式 $\delta::=T_f\leqslant c|T_f<c|T_f>c|T_f\geqslant c|\delta_1\wedge\delta_2$。

3)区间公式 $\varphi::=p|skip|(\varphi_1,\cdots,\varphi_m)\ prj\ \varphi_0|,(\varphi_1,\cdots,(\varphi_i,\cdots,\varphi_j)^{\Theta},\cdots,\varphi_m)\ prj\ \varphi_0|,\varphi_1;\varphi_2|\varphi*|\varphi_1\vee\varphi_2|\varphi_1\wedge\varphi_2|\varphi_1\parallel\varphi_2$。

4)时间公式 $\psi::=\varphi|\delta|\psi_1\wedge\psi_2|\psi_1\vee\psi_2|\psi_1;\psi_2$。

定义 5 RASL 导出公式定义如下：

1)$\bigcirc p::=skip;p$。

2)$more::=\bigcirc\text{true}$。

3)$empty::=\bigcirc\text{false}$。

4)$p;_{IQ}::=(p\wedge T_f\in I);q$。

定义 6(RASL 语义) 令 $c\in N$，令 $s_p(k)$ 表示 $p\in AP$ 在状态 s_k 下的真值。

RASL 满足关系被归纳定义如下：

1)$T=s_t(k)$。

2)$T_f=s_t(j)-s_t(k)$。

3)$I|=T_f\leqslant C$ 当且仅当 $s_t(j)-s_t(k)\leqslant C$。

4)$I|=T_f\geqslant C$ 当且仅当 $s_t(j)-s_t(k)\geqslant C$。

5)$I|=T_f<C$ 当且仅当 $s_t(j)-s_t(k)<C$。

6)$I|=T_f>C$ 当且仅当 $s_t(j)-s_t(k)>C$。

7)$I|=\delta_1\wedge\delta_2$ 当且仅当 $I|=\delta_1$ and $I|=\delta_2$。

8)$I|=p$ 当且仅当 $s_p(k)=\text{true}$。

9)$I|=\varphi_1\wedge\varphi_2$ 当且仅当 $I|=\varphi_1$ and $I|=\varphi_2$。

10)$I|=\varphi_1\vee\varphi_2$ 当且仅当 $I|=\varphi_1$ or $I|=\varphi_2$。

11)$I|=\varphi_1;\varphi_2$ 当且仅当 $\exists r,k\leqslant r\leqslant j$，使得 $(\sigma,i,k,r)|=\varphi_1$ 且 $(\sigma,i,k,r)|=\varphi_2$。

12)$I|=skip$ 当且仅当 $len(\sigma)=1$。

13)$I|=\varphi_1\parallel\varphi_2$，当且仅当 $\varphi_1\wedge(\varphi_2;\text{true})\vee\varphi_2\wedge(\varphi_1;\text{true})$。

14)$I|=\varphi^*$ 当且仅当：①存在无限多 $r_0,\cdots,r_n\in N_\omega$，使得 $k=r_0\leqslant r_1\leqslant\cdots\leqslant r_{n-1}\leqslant r_n=j$，$(\sigma,i,k,r_0)|=\varphi$，对于每一个 $1\leqslant l\leqslant n$，$(\sigma,r_{l-1},r_{l-1},r_l)|=\varphi$；②$k=j$。

15)$I-prj:I|=(\varphi_1,\cdots,\varphi_m)\ prj\ \psi$，当且仅当存在整数 $k=r_0\leqslant r_1\leqslant\cdots\leqslant r_m\leqslant j$，且 $(\sigma,i,k,r_1)|=\varphi_1,\cdots,(\sigma,r_{n-1},r_{n-1},r_n)|=\varphi_n$，其中 $1<n\leqslant m$，使得 σ' 在以下被提到的两种情境中，有 $(\sigma',0,0,|\sigma'|)|=\psi$：①$r_m<j$ 且 $\sigma'=\sigma\downarrow(r_0,\cdots,r_m)\cdot\sigma(r_m+1,\cdots,j)$；②$r_m=j$ and $\sigma'=\sigma\downarrow(r_0,\cdots,r_h)$，$0\leqslant h\leqslant m$。

16)$I|=(\varphi_1,\cdots,(\varphi_i,\cdots,\varphi_j)^{\Theta},\cdots,\varphi_m)\ prj\ \varphi_0$ 当且仅当存在 $n\in N_0$，使得 $I|=(\varphi_1,\cdots,(\varphi_i,\cdots,\varphi_j)^n,\cdots,\varphi_m)\ prj\ \varphi_0$。

17)$I|=\psi_1\wedge\psi_2$ 当且仅当 $I|=\psi_1$ 且 $I|=\psi_2$。

18)$I|=\psi_1\vee\psi_2$ 当且仅当 $I|=\psi_1$ 或 $I|=\psi_2$。

19)$I|=\psi_1;\psi_2$ 当且仅当 $\exists r,k\leqslant r\leqslant j$，使得 $(\sigma,i,k,r)|=\psi_1$ 且 $(\sigma,i,k,r)|=\psi_2$。

3 用时态逻辑公式构造攻击模型

3.1 back 攻击

3.1.1 攻击原理

back 攻击[15]是针对 Apache Web 服务器发起的，攻击者向服务器发送大量的 http 请求，这些请求包含大量的 URL 前字符(/)。当服务器试图处理这些请求时，它将无力去处理其他合法的请求，因此拒绝再向合法的用户提供正常的服务。

攻击的关键步骤可以描述为：1)Apache Web 服务器收到了来自攻击者的非法 http 请求；2)网络主机向 Apache Web 服务器发送合法的 http 请求；3)服务器收到合法的 http 请求；4)网络主机无法收到请求回复的数据包。

3.1.2 back 攻击的时序逻辑公式

根据 3.1.1 节中的攻击原理和关键步骤，为 back 攻击的原子动作定义了原子命题，如表 1 所列。

根据 back 攻击的攻击原理、关键步骤和原子命题之间的时序逻辑关系，使用命题区间时序逻辑(ITL)为 back 攻击建

立时序逻辑公式如下:

$$(ApacheWebserver.receive.http.p;ture)^* \rightarrow XF(\forall i(i.send.ApacheWebserver.http \wedge F(ApacheWebserver.receive.http_i \wedge G(\neg i.receive.ApacheWebserver)))) \quad (1)$$

表 1　back 攻击时序逻辑公式中的原子命题及其含义

Table 1　Atomic propositions in the formula for back attack and their meanings

原子命题	表示的含义
$ApacheWebserver.receive.http.p$	ApacheWeb 服务器收到了含有字段 p 的 http 请求
$i.send.ApacheWebserver.http$	网络主机 i 向 ApacheWeb 服务器发送 http 请求
$ApacheWebserver.receive.http_i$	ApacheWeb 服务器收到了网络主机 i 的 http 请求
$i.receive.ApacheWebserver$	网络主机 i 收到 ApacheWeb 服务器发回的数据包

3.2　ProcessTable 攻击

3.2.1　攻击原理

进程管理器(ProcessTable)攻击能够占据处于运行状态的操作系统的进程管理器,使系统运行缓慢甚至瘫痪,直到攻击消失或停止了进程攻击,这种现象才会消失。

对于进程管理器攻击,可以通过短期内记录被攻击者节点特殊端口的大量连接(TCP 连接)次数来进行检测。关键步骤可以描述为:1)以每个端口的连接作为初始状态,记录 2min 内连接的端口数量;2)针对某一特殊端口,记录 2min 内侵入节点端口的网络流量;3)在目标节点计算机上攻击者首次建立连接的端口数量大于 160,且每个端口的网络流量大于 9000bps。

3.2.2　ProcessTable 攻击的时序逻辑公式

根据 3.2.1 节中的攻击原理和关键步骤,为 ProcessTable 攻击的原子动作定义原子命题,如表 2 所列。

表 2　ProcessTable 攻击时序逻辑公式中的原子命题及其含义

Table 2　Atomic propositions in the formula for ProcessTable attack and their meanings

原子命题	表示的含义
p	开始计时时刻连接的端口数量为 0,所有端口的网络流量为 0
$attacked.port.connection.number$	被攻击者被建立连接的端口数量首次大于 160
$attacked.port_i.traffic$	被攻击者第 i 个端口的网络流量首次大于 9000bps(因为是 TCP 连接,所以 $0<i\leqslant 65535$)
x_1	开始计时到端口数量首次大于 160 的时间间隔
x_2	开始计时到出现某些端口网络流量首次大于 9000bps 的时间间隔

根据 ProcessTable 攻击的攻击原理、关键步骤和原子命题之间的时序逻辑关系,使用实时攻击签名逻辑(RASL)为 ProcessTable 攻击建立时序逻辑公式:

$$f=F((p \wedge Fattacked.port.connection.number \wedge (x_1<2)) \wedge (\exists i(p \wedge Fattacked.port_i,traffic \wedge (x_2<2)))) \quad (2)$$

其中,$0<i\leqslant 65535$。

3.3　Saint 攻击

3.3.1　攻击原理

Saint 是一个强大的系统漏洞扫描工具,它可以对目标网络或者主机进行安全漏洞的检测与分析,找出网络中安全隐患和存在的可能漏洞。该攻击的攻击原理是:首先探测目标系统中的活动主机,对活动主机进行端口扫描,确定其开放的端口,同时根据协议指纹技术识别主机的操作系统类型;然后,扫描器对开放的端口进行网络服务类型的识别,确定其提供的网络服务;最后,漏洞扫描器根据目标系统的操作系统平台和提供的网络服务,调用漏洞资料库中已知的各种漏洞进行逐一检测,通过对探测响应数据包的分析来判断是否存在已知安全漏洞。

Saint 扫描攻击主要分为 5 部分:主机发现,端口扫描,服务侦测,操作系统类型及版本侦测,系统漏洞检测。

1)主机发现

主机发现的关键步骤为:目标主机收到了攻击者发送的 ICMP 回应请求报文(ICMP echo request);目标主机 TCP 端口 443 收到了攻击者发送的 TCP SYN 数据包;目标主机 TCP 端口 80 收到了攻击者发送的 TCP ACK 数据包;目标主机收到了攻击者发送的 ICMP timestamp 时间戳请求。

2)端口扫描

端口扫描的主要目的是通过探测确定被攻击者系统开放的端口。下面分别介绍 8 种常见的端口扫描技术的原理及关键步骤。

①TCP Connect Scanning

原理:攻击者向目标主机发送 SYN 数据包,目标主机收到并返回 SYN/ACK 数据包,攻击者收到 SYN/ACK 数据包后返回 ACK 数据包,则说明目标主机服务的这个端口是开放的。

因此,TCP Connect Scanning 扫描技术分解的关键步骤为:目标主机收到了攻击者发送的 SYN 数据包;攻击者收到了目标主机返回的 SYN/ACK 数据包;目标主机收到了攻击者回复的 ACK 数据包。

②TCP SYN Scanning

原理:攻击者向目标主机发送自己构造的特殊 SYN 数据包,目标主机收到并返回 SYN/ACK 数据包,攻击者收到 SYN/ACK 数据包后得知目标主机服务的这个端口是开放的,但是攻击者却一直未向目标主机返回 ACK 数据包,目标主机也就永远无法收到攻击者回复的 ACK 数据包。

因此,TCP SYN Scanning 扫描技术分解的关键步骤为:目标主机收到了攻击者构造的特殊 SYN 数据包;攻击者收到了目标主机返回的 SYN/ACK 数据包;目标主机永远不能收到攻击者回复的 ACK 数据包。

③CP FIN Scanning

原理:攻击者向目标主机发送 FIN 数据包,如果目标主机不返回任何信息,则说明攻击者发送的 FIN 数据包到达的是一个开放的端口,攻击者也就永远不能收到目标主机返回的 RST 数据包。

因此,TCP FIN Scanning 扫描技术分解的关键步骤为:目标主机收到了攻击者发送的 FIN 数据包;攻击者永远不能收到目标主机返回的 RST 数据包。

④UDP Scanning

原理:攻击者向目标主机端口发送 UDP 数据包,目标主机如果端口是开放的,则不返回任何信息,攻击者也就不能收

到目标主机回复的 ICMP port unreachable 数据包。

因此,UDP Scanning 扫描技术分解的关键步骤为:目标主机收到了攻击者发送的 UDP 数据包;攻击者永远不能收到目标主机回复的 ICMP port unreachable 数据包。

⑤TCP ACK Scanning

原理:攻击者向目标主机发送 ACK 数据包,如果攻击者能够收到目标主机返回的 RST 数据包,则说明目标主机服务的这个端口没有被屏蔽。

因此,TCP ACK Scanning 扫描技术分解的关键步骤为:目标主机收到了攻击者发送的 ACK 数据包;攻击者收到了目标主机回复的 RST 数据包。

⑥TCP NULL Scanning

原理:攻击者将 TCP 数据包中的 ACK,FIN,RST,SYN,URG,PSH 位都置为空并发送给目标主机,如果目标主机服务的对应端口是开放的,则目标主机不返回任何信息,攻击者也就永远不能收到 RST 数据包。

因此,TCP NULL Scanning 扫描技术分解的关键步骤为:目标主机收到了攻击者发送的 ACK,FIN,RST,SYN,URG,PSH 位置都为空的 TCP 数据包;攻击者永远不能收到目标主机回复的 RST 数据包。

⑦TCP XMAX Scanning

原理:攻击者将 TCP 数据包中的 ACK,FIN,RST,SYN,URG,PSH 位都置为 1 后发送给目标主机,如果目标主机服务的对应端口是开放的,则目标主机不返回任何信息,攻击者也就永远不能收到 RST 数据包。

因此,TCP XMAX Scanning 扫描技术分解的关键步骤为:目标主机收到了攻击者发送的 ACK,FIN,RST,SYN,URG,PSH 位置都为 1 的 TCP 数据包;攻击者永远不能收到目标主机回复的 RST 数据包。

⑧SYN/ACK Scanning

原理:攻击者向目标主机发送 SYN/ACK 数据包,如果目标主机服务的对应端口是开放的,则目标主机不返回任何信息,攻击者也就永远不能收到 RST 数据包。

因此,SYN/ACK Scanning 扫描分解的关键步骤为:目标主机收到了攻击者发送的 SYN/ACK 数据包;攻击者永远不能收到目标主机回复的 RST 数据包。

3)运行服务类型及版本侦测

攻击者对使用 TCP/IP 网络服务协议中的任意一种协议 i 对步骤 2)中扫描出目标主机上开放端口的任意一个端口 j 进行探测,如果能得到目标主机开放端口 j 上的正确响应,则可以确认 j 上的服务就是 i 服务,同时根据目标主机返回数据包的不同确定服务的版本;否则继续使用其他 TCP/IP 网络服务协议进行探测,直到探测成功(也就是说,攻击者一直向目标主机的开放端口发送 TCP/IP 网络服务协议的数据包,直到攻击者收到了目标主机正确响应返回的数据包为止)。

因此,运行服务类型及版本侦测分解的关键步骤为:攻击者向目标主机的开放端口 j 发送 TCP/IP 网络服务协议 i 的数据包;攻击者收到了目标主机的开放端口 j 正确回应协议 i 的数据包。

4)操作系统类型及版本侦测

攻击者收集来自目标主机的 TCP/IP 数据包,提取出 IP 包首部中的 TTL 和 DF 位,以及 TCP 包中的 Window 和 TOS 位,再对照已经建立好的 TCP/IP 栈特征数据库,查找出(或者以概率的形式给出)目标主机的操作系统类型及版本。

因此,操作系统类型及版本侦测分解的关键步骤为:攻击者收集来自网络目标主机的 TCP/IP 数据包;提取目标主机 TCP 数据包中的 Window 和 TOS 位及 IP 包首部中的 TTL 和 DF 位;对照已经建立好的 TCP/IP 栈特征数据库,查找出目标主机的操作系统类型及版本。

5)系统中漏洞的检测

从步骤 2)中攻击者已经获取的目标主机上开放的服务端口中挑选出想要探测漏洞的那些服务端口,并向目标主机发送精心构造的特殊通讯 TCP 数据包,目标主机返回数据包(不同版本的服务和操作系统返回不同版本的数据包),攻击者通过分析目标主机返回的数据包的特殊片段来判断漏洞是否存在(主要判断漏洞是否被修补)。

因此,操作系统类型及版本侦测技术分解的关键步骤为:目标主机收到了来自攻击者精心构造的特殊通讯的 TCP 数据包;攻击者收到了来自目标主机返回的数据包。

3.3.2 Saint 攻击的时序逻辑公式

根据 3.3.1 节中的攻击原理和关键步骤,分别为 Saint 攻击的 5 个部分定义了原子命题,并使用命题区间时序逻辑(ITL)为各部分建立时序逻辑公式。

1)发现目标主机

发现目标主机的时序逻辑公式可描述为:

$$\varphi_1 = targethosts.receive.ICMP\text{-}echo\text{-}request \vee targethosts_{(port=443)}.receive.TCPSYN \vee targethosts_{(port=80)}.receive.TCPACK \vee targethosts.receive.ICMP\text{-}timestamp\text{-}request$$

其中,$targethosts.receive.ICMP\text{-}echo\text{-}request$ 表示目标主机收到了攻击者发送的 ICMP 回应请求报文 ICMP echo reques;$targethosts_{(port=443)}.receive.TCPSYN$ 表示目标主机 TCP 端口 443 收到了攻击者发送的 TCP SYN 数据包;$targethosts_{(port=80)}.receive.TCPACK$ 表示目标主机 TCP 端口 80 收到了攻击者发送的 ACK 数据包;$targethosts.receive.ICMP\text{-}timestamp\text{-}request$ 表示目标主机收到了攻击者发送的 ICMP 时间戳请求报文 ICMP timestamp request。

2)端口扫描:

①TCP Connect scanning

$$[f_1 = G\,targethosts.receive.attack.SYN \wedge F(attack.receive.targethosts.SYNACK \wedge F targethosts.receive.attack.ACK)]$$

$targethosts.receive.attack.SYN$ 表示目标主机收到了攻击者发送的 SYN 数据包;$attack.receive.targethosts.SYNACK$ 表示攻击者收到了目标主机返回的 SYN/ACK 数据包;$targethosts.receive.attack.ACK$ 表示目标主机收到了攻击者回复的 ACK 数据包。

②TCP SYN Scanning

$$f_2 = G[targethosts.receive.attack.special.SYN \wedge F(attack.receive.targethosts.SYNACK \wedge G(\neg targethosts.receive.attack.ACK))]$$

$targethosts.receive.attack.special.SYN$ 表示目标主机收到

了攻击者自身构造的特殊的 SYN 数据包;*attack. receive. targethosts. SYNACK* 表示攻击者收到了目标主机返回的 SYN/ACK 数据包;*targethosts. receive. attack. ACK* 表示目标主机收到了攻击者回复的 ACK 数据包。

③TCP FIN Scanning

$f_3 = G[targethosts.\ receive.\ attack.\ FIN \wedge G(\neg attack.\ receive.\ targethosts.\ RST)]$

targethosts. receive. attack. FIN 表示目标主机收到了攻击者发送的 FIN 数据包;*attack. receive. targethosts. RST* 表示攻击者收到了目标主机返回的 RST 数据包。

④UDP Scanning

$f_4 = G[targethosts.\ receive.\ attack.\ UDP \wedge G(\neg attack.\ receive.\ targethosts.\ ICMP portunreachable)]$

targethosts. receive. attack. UDP 表示目标主机收到了攻击者发送的 UDP 数据包;*attack. receive. targethosts. ICMP portunreachable* 表示攻击者收到了目标主机返回的 ICMP port unreachable 数据包。

⑤TCP ACK scanning

$f_5 = G(targethosts.\ receive.\ attack.\ ACK \wedge F\ attack.\ receive.\ targethosts.\ RST)$

targethosts. receive. attack. ACK 表示目标主机收到了攻击者发送的 UDP 数据包;*attack. receive. targethosts. RST* 表示攻击者收到了目标主机返回的 ICMP port unreachable 数据包。

⑥TCP NULL Scanning

$f_6 = G[targethosts.\ receive.\ attack.\ TCP_{(ACK=0,FIN=0,RST=0,SYN=0,URG=0,PSH=0)} \wedge G(\neg attack.\ receive.\ targethosts.\ RST)]$

targethosts. receive. attack. $TCP_{(ACK=0,FIN=0,RST=0,SYN=0,URG=0,PSH=0)}$ 表示目标主机收到了攻击者发送的 ACK,FIN,RST,SYN,URG,PSH 位置都为空的 TCP 数据包;*attack. receive. targethosts. RST* 表示攻击者收到了目标主机返回的 RST 数据包。

⑦TCP XMAX Scanning

$f_7 = G[targethosts.\ receive.\ attack.\ TCP_{(ACK=1,FIN=1,RST=1,SYN=1,URG=1,PSH=1)} \wedge G(\neg attack.\ receive.\ targethosts.\ RST)]$

targethosts. receive. attack. $TCP_{(ACK=1,FIN=1,RST=1,SYN=1,URG=1,PSH=1)}$ 表示目标主机收到了攻击者发送的 ACK,FIN,RST,SYN,URG,PSH 位置都为 1 的 TCP 数据包;*attack. receive. targethosts. RST* 表示攻击者收到了目标主机返回的 RST 数据包。

⑧SYN/ACK Scanning

$f_8 = G[targethosts.\ receive.\ attack.\ SYNACK \wedge G(\neg attack.\ receive.\ targethosts.\ RST)]$

targethosts. receive. attack. SYNACK 表示目标主机收到了攻击者发送的 SYN/ACK 数据包;*attack. receive. targethosts. RST* 表示攻击者收到了目标主机返回的 RST 数据包。

由此可以得到,端口扫描的时序逻辑公式为:

$$\varphi_2 = f_1 \vee f_2 \vee f_3 \vee f_4 \vee f_5 \vee f_6 \vee f_7 \vee f_8$$
$$= G[targethosts.\ receive.\ attack.\ SYN \wedge F(attack.\ receive.\ targethosts.\ SYNACK \wedge F targethosts.\ receive.\ attack.\ ACK)] \vee G[targethosts.\ receive.\ attack.\ special.\ SYN \wedge F(attack.\ receive.\ targethosts.\ SYNACK \wedge G(\neg targethosts.\ receive.\ attack.\ ACK))] \vee G[targethosts.\ receive.\ attack.\ FIN \wedge G(\neg attack.\ receive.\ targethosts.\ RST)] \vee G[targethosts.\ receive.\ attack.\ UDP \wedge G(\neg attack.\ receive.\ targethosts.\ ICMP portunreachable)] \vee G(targethosts.\ receive.\ attack.\ ACK \wedge F attack.\ receive.\ targethosts.\ RST) \vee G[targethosts.\ receive.\ attack.\ TCP_{(ACK=0,FIN=0,RST=0,SYN=0,URG=0,PSH=0)} \wedge G(\neg attack.\ receive.\ targethosts.\ RST)] \vee G[targethosts.\ receive.\ attack.\ TCP_{(ACK=1,FIN=1,RST=1,SYN=1,URG=1,PSH=1)} \wedge G(\neg attack.\ receive.\ targethosts.\ RST)] \vee G[targethosts.\ receive.\ attack.\ SYNACK \wedge G(\neg attack.\ receive.\ targethosts.\ RST)]$$

3)运行服务类型及版本侦测

运行服务类型及版本侦测的时序逻辑公式可描述为:

$\varphi_3 = G[(attack.\ send.\ TCPIP protocol_i.\ totargethosts port_j; \text{true})^* W(attack.\ receive.\ targethosts port_j.\ reply_i)]$

其中,$TCPIP protocol_i$ 表示 TCP/IP 网络服务协议的任意一种(用 i 区分);$targethosts port_j$ 表示目标主机开放端口的任意一个(用 j 区分);$attack. send. TCPIP protocol_i. totargethosts port_j$ 表示攻击者向目标主机的开放端口 j 发送 TCP/IP 网络服务协议 i 的数据包;$attack. receive. targethosts port_j. reply_i$ 表示攻击者收到了目标主机的开放端口 j 正确回应协议 i 的数据包。

4)操作系统类型及版本侦测

运行服务类型及版本侦测的时序逻辑公式可描述为:

$\varphi_4 = G[(attack.\ collect.\ targethosts.\ TCPIP; \text{true})^* \rightarrow F(((\exists TTL.\ IP.\ targethosts = TTL.\ IP.\ SignatureDatabase) \wedge (\exists DF.\ IP.\ targethosts = DF.\ IP.\ SignatureDatabase)) \wedge ((\exists Window.\ TCP.\ targethosts = Window.\ TCP.\ SignatureDatabase) \wedge (\exists TOS.\ TCP.\ targethosts = TOS.\ TCP.\ SignatureDatabase)))]$

其中,*attack. collect. targethosts. TCPIP* 表示攻击者收集来自网络目标主机的 TCP/IP 数据包;*TTL. IP. targethosts* 表示目标主机 IP 包首部中的 TTL 位;*DF. IP. targethosts* 表示目标主机 IP 包首部中的 DF 位;*Window. TCP. targethosts* 表示目标主机 TCP 包中的 Window 位;*TOS. TCP. targethosts* 表示目标主机 TCP 包中的 TOS 位;*TTL. IP. SignatureDatabase* 表示 TCP/IP 栈特征数据库中 IP 包首部中的 TTL 位;*DF. IP. SignatureDatabase* 表示 TCP/IP 栈特征数据库中 IP 包首部中的 DF 位;*Window. TCP. SignatureDatabase* 表示 TCP/IP 栈特征数据库中 TCP 包中的 Window 位;*TOS. TCP. SignatureDatabase* 表示 TCP/IP 栈特征数据库中 TCP 包中的 TOS 位。

5)系统漏洞检测

系统漏洞检测的时序逻辑公式可描述为:

$\varphi_5 = G(targethosts.\ receive.\ attack.\ special.\ TCP \wedge Fattack.\ receive.\ targethosts.\ reply)$

其中,*targethosts. receive. attack. special. TCP* 表示目标主机收到了来自攻击者精心构造的特殊通讯的 TCP 数据包;

attack.*receive*.*targethosts*.*reply*表示攻击者收到了来自目标主机返回的数据包。

最后，根据 Saint 攻击的 5 个部分之间的时序逻辑关系，给出 Saint 攻击的时序逻辑公式：

$$\varphi=G(\varphi_1 \wedge F(\varphi_2 \wedge F(\varphi_3 \wedge \varphi_4 \wedge F\varphi_5))) \quad (3)$$

4 3 种攻击的入侵检测方法

基于式(1)和式(3)，给出基于命题区间时序逻辑(ITL)模型检测的入侵检测方法来检测 back 攻击和 Saint 攻击。新检测方法的原理如图 1 所示。

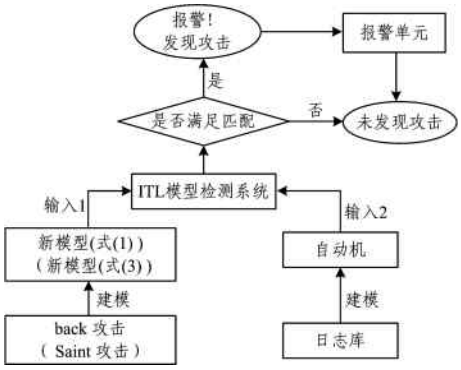

图 1 基于 ITL 的模型检测方法检测 back 攻击和 Saint 攻击的原理

Fig. 1 Model checking method for detecting back attack and Saint attack based on ITL

基于式(2)，给出基于实时攻击签名逻辑(RASL)模型检测的入侵检测方法来检测 ProcessTable 攻击。新检测方法的原理如图 2 所示。

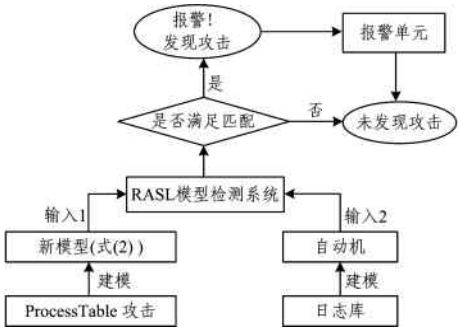

图 2 基于 RASL 的模型检测方法检测 ProcessTable 攻击的原理

Fig. 2 Model checking method for detecting ProcessTable attack based on RASL

结束语 本文的主要成果是使用命题区间时序逻辑(ITL)和实时攻击签名逻辑(RASL)分别为常见的 back 攻击、Saint 攻击和 ProcessTable 攻击建立了时序逻辑公式模型，并分别给出了 3 种基于模型检测的入侵检测方法来检测这 3 种攻击，扩大了基于模型检测的入侵检测方法的适用范围。

参 考 文 献

[1] RPGER M, GOUBAULT-LARRECQ J. Log Auditing through Model-Checking [C]// IEEE Workshop on Computer Security Foundations. 2001: 220-234.

[2] SUN Y, WU T Y, MA X Q, et al. Modeling and verifying EPC network intrusion system based on timed automata[J]. Pervasive and Mobile Computing, 2016, 24(C): 61-67.

[3] CHAKIR E M, KHAMLICHI Y I, MOUGHIT M. Handling alerts for intrusion detection system using stateful pattern matching[C]// IEEE 4th International Colloquium on Information Science and Technology. 2017: 139-144.

[4] CLARKE E M, GRUNBERG O, PELED D A. Model Checking [M]. MIT Press Cambridge, MA, USA, 1997: 54-56.

[5] RAMPURE V, TIWARI A. A rough set based feature selection on KDD CUP 99 data set[J]. International Journal of Database Theory and Application, 2015, 8(1): 149-156.

[6] CHEN X Y. Research on Network Attack Model Based on Temporal Logic [D]. Zhengzhou: Zhengzhou University, 2014. (in Chinese)
陈茜月. 基于时序逻辑的网络攻击建模研究[D]. 郑州：郑州大学，2014.

[7] HU W. Modeling and Detection of Network Attack Based on Temporal Logic [D]. Zhengzhou: Zhengzhou University, 2015. (in Chinese)
胡伟. 基于时序逻辑的网络攻击建模与检测[D]. 郑州：郑州大学，2015.

[8] XU P T. Intrusion Detection Based On Model Checking Benchmark Test Platform For Research [D]. Zhengzhou: Zhengzhou University, 2016. (in Chinese)
许鹏涛. 基于模型检测的入侵检测基准测试平台研究[D]. 郑州：郑州大学，2016.

[9] ZHU W J, WANG Z Y, ZHANG H B. A novel algorithm for Intrusion Detection based on Model Checking Interval Temporal Logic [J]. China Communications, 2011, 8(3): 66-72. (in Chinese)
朱维军，王忠勇，张海滨. 一种基于区间时序逻辑模型检测的入侵检测算法[J]. 中国通信，2011，8(3)：66-72.

[10] DUAN Z H, TIAN C, ZHANG L. A decision procedure for propositional projection temporal logic with infinite models [J]. Acta Informatica, 2008, 45(1): 43-78.

[11] TIAN C, DUAN Z H. Complexity of propositional projection temporal logic with star [J]. Mathematical Structures in Computer Science, 2009, 19(1): 73-100.

[12] ZHU W J. Model Checking Timed Interval Temporal Logic: Theory, Algorithms and Application[D]. Xi'an: Xidian University, 2011. (in Chinese)
朱维军. 时间区间时序逻辑模型检测：理论、算法及应用[D]. 西安：西安电子科技大学，2011.

[13] ZHU W J, ZHOU Q L, YANG W D, et al. A Novel Algorithm for Intrusion Detection Based on RASL Model Checking [J]. Mathematical Problems in Engineering, 2013, 2013(3): 1-10. (in Chinese)
朱维军，周清雷，杨卫东，等. 一种基于实时攻击签名逻辑模型检测的入侵检测算法[J]. 工程中的数学问题，2013，2013(3)：1-10.

[14] ZHU W J, ZHANG H B, ZHOU Q L. On the Decidability of Satisfiability of Discrete TITL formulae [J]. Acta Electronica Sinica, 2010, 38(5): 1039-1045. (in Chinese)
朱维军，张海滨，周清雷. 离散时间区间时序逻辑可满足性的判定[J]. 电子学报，2010，38(5)：1039-1045.

[15] SOLANKAR P, PINGALE S. International Journal of Computer Science and Information Technologies[J]. International Journal of Computer Science and Information Technologies (IJCSIT), 2015, 6(2): 1096-1099.