(1) 对于 Creation 状态,在漏洞产生后的第 1 天时,其发生概率为 1,随着时间的推进,漏洞状态向 Discovery 逐渐演化,因此 Creation 状态的概率不断降低,在前 20 天内,曲线的斜率较陡,说明概率下降的速度快,到第 6 天时,概率值已经降低为 0.531 4,到第 15 天时,概率值降为 0.205 9,说明漏洞维持在该状态的可能性较低,会迅速向后续状态演化.

(2) 漏洞初始处于 Disclosure 状态的概率为 0,到第 3 天时,概率值陡增至 0.03,说明在漏洞产生之后,会迅速进入待公开状态,之后随着时间的推进,漏洞开始由 Disclosure 状态向 Exploit 和 Patched 状态的过渡,此时 Disclosure 状态的概率值不断降低;同时,当漏洞到达稳定的 Exploit 或者 Patched 状态时,相对应时刻的 Disclosure 概率值为 0.

(3) 对于 Exploit 和 Patched 状态,初始概率值为 0,随着时间的推进,漏洞逐步被发现和披露,漏洞利用不断扩散,Exploit 状态的发生概率不断增大.由于软件厂商努力研发修复补丁,加紧开发进度,因此漏洞演化到 Patched 的概率也不断增加.可以看出,Exploit 概率值增速要小于 Patched 的增速.例如第 8 天时,被利用的概率为 0.212 7,补丁发布的概率为 0.293 0,说明此时漏洞被修复的概率要始终高于被利用的概率.通过调节参数可以改变漏洞的时间概率,为安全漏洞防御提供指导.

(4) 由表 4 可以看出,在任意时间,漏洞处于不同状态的概率之和为 1;同时,随着时间的推进,最终漏洞演化到稳定的吸收态,即 Exploit 的概率为 0.42,Patched 的概率为 0.58,概率之和为 1.

2) 通过调节参数λ_1~λ_6 可以分析参数变化对各漏洞状态时间概率的影响规律,由于涉及变量多,为了简化讨论,考虑其他参数值不变,仅以调节λ_1 为例说明.利用第 3.1 节方法 1 计算λ_1=0.1,0.3,0.5,0.7,0.9 这 5 种情况下的状态时间概率,并绘制概率值在时间维的分布,如图 6 所示.

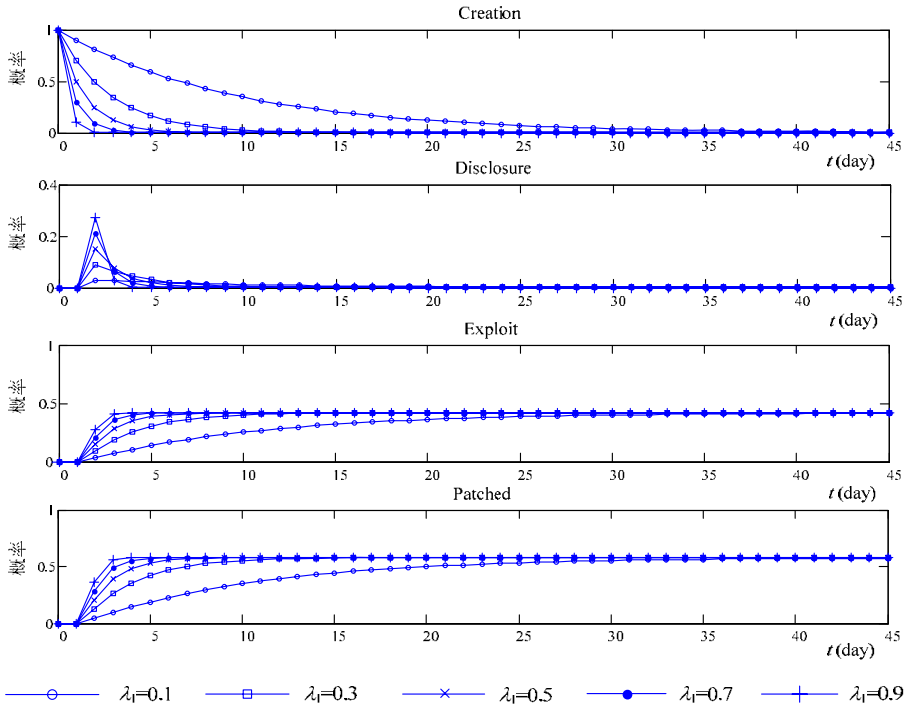

Fig.6 Time probability of each state in vulnerability life cycle with different λ_1

图 6 不同λ_1下的漏洞生命周期状态时间概率

从图 6 可以看出,参数λ_1的取值对各状态发生概率在时间维的走势影响不大;对于所有状态,当λ_1的取值更大时,其发生概率的变化更迅速,能够更快地趋向于稳定状态.上述发现对于研究漏洞生命周期的时间长度具有

重要意义,例如,当新增漏洞的产生速度加快时,宏观上可以判定多数漏洞的生命周期在缩短,漏洞在较短时间内能够演化到稳定状态,这意味着攻击者利用漏洞的周期也在缩短,对于指导软件厂家和安全漏洞研究机构加快推进补丁研发速度、缩短研发周期具有参考价值.通过调节其他参数也可以得到一些有益的结果,此处不再赘述.

4.3 漏洞利用及修复的期望概率值计算

接着第 4.1 节,首先考虑一般意义上的单个漏洞.

由第 3.2 节方法 2 可以计算出期望概率矩阵 $B=\begin{bmatrix}\lambda_2+\lambda_3\times\lambda_5 & \lambda_4+\lambda_3\times\lambda_6\\ \lambda_2+\lambda_3\times\lambda_5 & \lambda_4+\lambda_3\times\lambda_6\\ \lambda_5 & \lambda_6\end{bmatrix}$,则漏洞被利用的期望概率值是 $\lambda_2+\lambda_3\cdot\lambda_5$,被修复的期望概率值是 $\lambda_4+\lambda_3\cdot\lambda_6$.当 λ_1=0.1,λ_2=0.3,λ_3=0.3,λ_4=0.4,λ_5=0.4,λ_6=0.6 时,漏洞被利用的期望概率 0.42,被修复的期望概率 0.58,与第 4.1 节表 4 得出的稳定状态(t=94 天)概率结果一致,与预期相符;同时,说明此时漏洞生命周期为 93 天.

定义漏洞被利用和修复的期望概率差值 $f=|(\lambda_2-\lambda_4)+\lambda_3\times(\lambda_5-\lambda_6)|$,显然,当 $\lambda_2-\lambda_4$ 和 $\lambda_5-\lambda_6$ 不变时,λ_3 越大,漏洞被利用和修复的概率之差越大,表明提高已“发现”漏洞被“公开”的概率,能够同时提高漏洞被利用或修复的可能性.对于实际应用的指导意义是:当软件厂商的补丁研发水平提高时,应提高对发现漏洞的公开概率,此举能够从整体上提高漏洞被修复的概率,当软件研发能力受限时,应降低对发现漏洞的公开概率,避免利用方法扩散,导致漏洞被更容易演化到 Exploit 状态.

4.4 安全漏洞的时间维度风险分析

结合图 4,本节分析遭受 WannaCry 入侵的系统在时间维度上的风险变化趋势.由表 3 可知,WannaCry 攻击利用漏洞均来自厂商 Microsoft,查询 CVE 数据库可知,截至 09/10/2017,供应商 Microsoft 的发布的漏洞总数 $Num(Microsoft,Disco)_1$=4236,近 k=1 年来(09/10/2016~09/10/2017)发布的不同 Level 漏洞的历史统计数据见表 5;然后,利用第 2 节表 2 计算不同 Level 漏洞在其生命周期中的状态转移概率,见表 6.

Table 5 CVE history statistics information of Microsoft vulnerabilities

表 5 Microsoft 漏洞的 CVE 历史数据统计信息

漏洞数量统计	*Level*=Low	*Level*=Medium	*Level*=High
$Num(Microsoft,Level,Disco)_1$	1 742	1 342	1 152
$Num(Microsoft,Level,Discl)_1$	1 537	1 037	737
$NumAdd(Microsoft,Level,Disco)_1$	479	385	398
$Num(Microsoft,Level,(Disco\rightarrow Explo))_1$	28	122	170
$Num(Microsoft,Level,(Disco\rightarrow Patch))_1$	1 390	1 081	913
$Num(Microsoft,Level,(Discl\rightarrow Patch))_1$	606	409	285

Table 6 Calculations of transition probabilities for different vulnerability levels of Microsoft

表 6 Microsoft 不同 Level 漏洞在其生命周期内的状态转移概率计算结果

Level	λ_1	λ_2	λ_3	λ_4	λ_5	λ_6
Low	0.275 0	0.016 1	0.186 0	0.797 9	0.605 7	0.394 3
Medium	0.286 9	0.090 9	0.103 6	0.805 5	0.605 6	0.394 4
High	0.345 5	0.147 6	0.059 9	0.792 5	0.613 3	0.386 7

可以看出,不同 Level 漏洞的生命周期时间模型中的状态转移概率不同.接着,利用第 3.1 节方法 1 计算厂商 Microsoft 不同 Level 漏洞处于生命周期 Exploit 状态的时间概率,记录在表 7 中,并绘制概率值在时间维度上的分布如图 7 所示.

从图 7 和表 7 可以看出,对于利用难度 High 的漏洞,稳定后到达 Exploit 状态的期望概率为 0.128 7,漏洞生命周期为 32 天,对应图 7 中的点 *C*;难度 Medium 的漏洞,Exploit 状态的期望概率为 0.153 6,漏洞生命周期为 26 天,对应图 7 中的点 *B*;难度 Low 的漏洞,Exploit 状态的期望概率为 0.184 3,生命周期为 22 天,在图中用 *A* 标出.

Table 7 Probabilities of Exploit for different vulnerability levels of Microsoft in their life cycle

表 7 Microsoft 不同 Level 漏洞在其生命周期内的 Exploit 概率

Day/Level	Low	Medium	High	Day/Level	Low	Medium	High
1	0	0	0	2	0.004 4	0	0.004 4
4	0.097 1	0.062 7	0.063 4	6	0.147 0	0.107 4	0.094 4
8	0.168 3	0.130 1	0.110 7	10	0.177 5	0.141 7	0.119 3
12	0.181 4	0.147 6	0.123 8	14	0.183 1	0.150 5	0.126 1
16	0.183 8	0.152 1	0.127 4	18	0.184 1	0.152 8	0.128 0
20	0.184 2	0.153 2	0.128 4	**22**	**0.184 3**	0.153 4	0.128 6
24	0.184 3	0.153 5	0.128 7	**26**	0.184 3	**0.153 6**	0.128 7
28	0.184 3	0.153 6	0.128 7	30	0.184 3	0.153 6	0.128 7
32	0.184 3	0.153 6	**0.128 8**	34	0.184 3	0.153 6	0.128 8
36	0.184 3	0.153 6	0.128 8	38	0.184 3	0.153 6	0.128 8
…	…	…	…	…	…	…	…

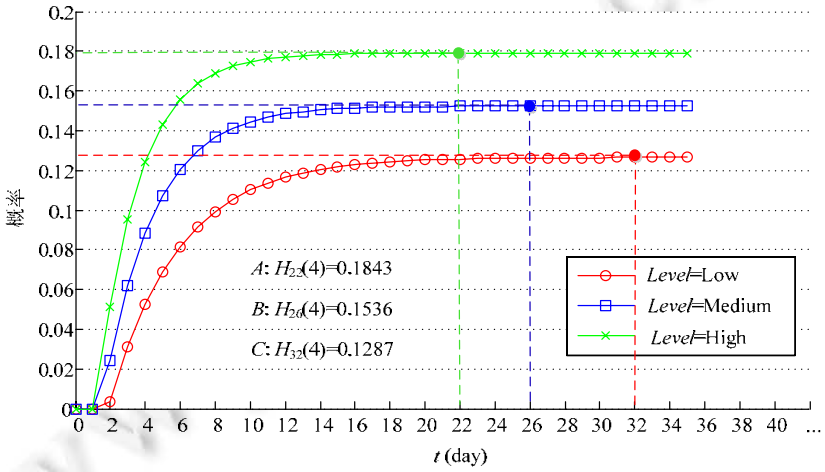

Fig.7 Probability distributions of Exploit for different vulnerability levels of Microsoft in their life cycle

图 7 Microsoft 不同 Level 漏洞在生命周期内的 Exploit 概率分布

上述结果从宏观角度预测了未来一段时间内,Microsoft 产品不同 Level 漏洞被利用的平均概率随时间的变化规律.不难看出,在同一时刻,难度为 Low 的漏洞被利用的概率最高,而难度为 High 的漏洞被利用的概率最低,但后者到达稳定状态所需时间更长.该结论与实际漏洞利用攻击相符,对于难度低的漏洞,随着攻击脚本的传播,低水平的攻击者也能成功发动攻击,导致漏洞被利用概率增大;并且随着攻击方法的迅速传播,漏洞状态能够更快地趋于稳定,因此生命周期更短.

通过上述分析,对于新披露的漏洞,通过查询相应厂商的历史漏洞信息,并结合 CVSS 评估漏洞利用难度 Level,利用本文方法可以预测漏洞生命周期长度以及时间维度的利用概率,预测结果从整体上为安全漏洞的风险控制提供了参考.

下面分析图 4 网络系统在漏洞生命周期内的风险变化情况.对于 WannaCry 攻击利用的 2 个漏洞,结合表 3 中的漏洞基础信息,以漏洞 CVE-2017-7494 的公开时间(04/05/2017)为起始时间,设定该时刻为 t=0,然后利用第 3.3 节方法 3,计算被入侵系统中的单个漏洞及系统整体风险值,记录在表 8 中,并绘制漏洞时间风险变化如图 8 所示.由图 8 可知,单个漏洞和整体系统的风险值在时间维度上均符合指数分布.在初始阶段,$Vuln_1$ 会产生安全风险,而 $Vuln_2$ 不存在风险隐患,System 风险处于较低水平.随着 $Vuln_2$ 的公开,WannaCry 蠕虫开始利用此漏洞搜寻入侵目标主机,并结合 $Vuln_1$ 实现在局域网内横向渗透.随后几天内,受感染主机数量急剧增多,因此 System 风险迅速上升.由于 $Vuln_2$ 的 *ExpSco* 高于 $Vuln_1$,因此前者的单漏洞风险值更高.结合表 8 可知:在 t=32 附近,System 风险达到一个较高的稳定值,表现为勒索病毒在全球范围大爆发,与著名软件供应商 Symante[28]提供的勒索病

毒集中爆发时间为 12/05/2017 较为接近,验证了本文方法的有效性和准确性.

Table 8 Time risk values of WannaCry blackmail attacks

表 8 WannaCry 勒索攻击的时间风险值

t (day)	$Vuln_1$	$Vuln_2$	System	t (day)	$Vuln_1$	$Vuln_2$	System	t(Day)	$Vuln_1$	$Vuln_2$	System
0	0	0	0	2	0	0.889	0.444	4	0	1.873	0.936
6	0	2.390	1.195	8	0.047	2.662	1.354	10	0.673	2.804	1.739
12	1.003	2.880	1.941	14	1.176	2.919	2.048	16	1.267	2.940	2.103
18	1.315	2.951	2.133	20	1.340	2.956	2.148	22	1.353	2.960	2.156
24	1.360	2.961	2.160	26	1.363	2.962	2.163	28	1.366	2.962	2.164
30	1.366	2.963	2.164	**32**	**1.367**	**2.963**	**2.165**	34	1.367	2.963	2.165
…	…	…	…	…	…	…	…	…	…	…	…

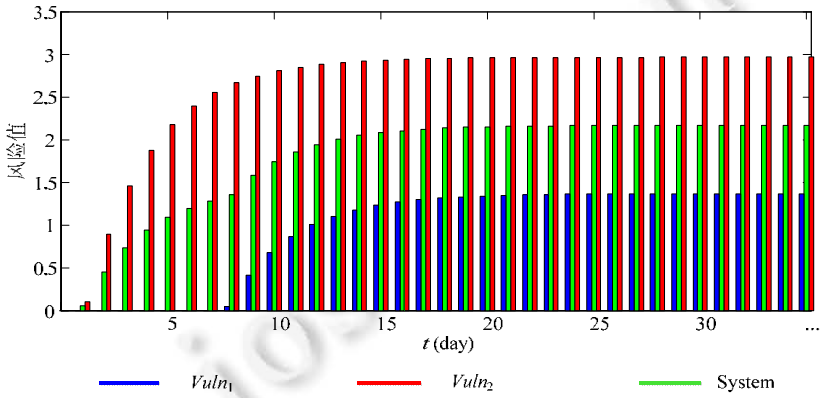

Fig.8 Security risk trend of WannaCry blackmail attacks

图 8 WannaCry 勒索攻击的安全风险趋势

4.5 方案综合比较

本文与典型相关研究的特点比较见表 9,从表中可以看出,文献[11]通过对漏洞威胁和利用可能性划分等级,结合风险矩阵计算系统的安全风险等级,但等级划分具有较强的主观性;文献[12]结合漏洞属性评估漏洞影响度,该过程依赖领域知识,分析未知漏洞的能力不足;文献[15]基于层次分析法,但系统安全指标体系难以建立,且未对风险结果进行量化;文献[13]结合粗集理论实现了漏洞属性约简,但只能分析已知漏洞风险;文献[14]结合先验的漏洞数据库,通过 CVSS 数据集训练贝叶斯信念网络,在某种程度上能够度量未知漏洞,且结果的客观性较高.但以上研究均未结合漏洞生命周期过程,缺乏时间维度上的动态度量.相比之下,本文结合漏洞生命周期过程,从宏观角度分析了漏洞状态的一般演化规律,度量安全漏洞的风险水平,具备对未知漏洞风险预测的能力,给出了漏洞利用概率在时间维度上的变化函数,能够实时、动态地度量网络风险,更真实地反映系统安全漏洞的风险状况,为安全管理员的风险管理提供更科学的决策支持.

Table 9 Comprehensive comparisons among our method and others

表 9 本文方法与其他方法综合比较

类型	客观程度	未知漏洞	动态度量	风险值量化	时间维分析
文献[11]	低	否	否	否	否
文献[12]	低	否	否	是	否
文献[13]	中	否	否	否	否
文献[14]	高	是	否	是	否
文献[15]	低	否	否	否	否
本文方法	中	是	是	是	是

5 结束语

信息系统安全漏洞是系统设计本身的缺陷,它可能引起系统异常、瘫痪,甚至被黑客利用进行入侵,引起更大的损失,因此,安全漏洞风险评估是信息系统安全研究的重要内容.目前,关于安全漏洞风险研究大多的静态的分析方法,未考虑漏洞利用概率在漏洞产生时间维度上的分布规律,缺少对于漏洞生命周期整体的考虑.

本文从时间维度出发,利用吸收 Markov 链对漏洞生命周期中的状态迁移进行建模,以 CVE 的漏洞数据库为输入,依据历史先验漏洞信息构造状态转移概率矩阵,通过矩阵推导,在时间维上对安全风险进行量化分析,真实、动态地呈现漏洞的时间风险.实例分析结果表明,文中提出的模型及方法符合实际应用,结果准确、有效.

References:

[1] Chen K, Feng DG, Su PE, Nie CJ, Zhang XF. Multi-Cycle vulnerability discovery model for prediction. Ruan Jian Xue Bao/Journal of Software, 2010,21(9):2367−2375 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/3626.htm [doi: 10.3724/SP.J.1001.2010.03626]

[2] Nie CJ, Zhao XF, Chen K, Han ZQ. An software vulnerability number prediction model based on micro-parameters. Chinese Journal of Computer Research & Development, 2011,48(7):1279−1287 (in Chinese with English abstract).

[3] Gao ZW, Yao Y, Rao F, Liu YZ, Luo P. Predicting model of vulnerabilities based on the type of vulnerability security. Chinese Acta Electronica Sinica, 2013,41(9):1784−1787 (in Chinese with English abstract).

[4] Arbaugh WA, Fithen WL, Mchugh J. Windows of vulnerability: A case study analysis. Computer, 2000,33(12):52−59. [doi: 10.1109/2.889093]

[5] Frei S. Security econometrics: The dynamics of (in) security. [Ph.D. Thesis]. Sissach: ETH Zurich, 2009.

[6] Kaaniche M, Marconato GV, Nicomette V. Security-Related vulnerability life cycle analysis. In: Proc. of the 2013 IEEE Int'l Conf. on Risk and Security of Internet & Systems. 2013. 1−8. [doi: 10.1109/CRISIS.2012.6378954]

[7] Song MQ, Wang LL, Yu B. Research on time risk of security vulnerability based on lifecycle theory. Chinese Journal of Computer Engineering, 2011,37(1):131−133 (in Chinese with English abstract).

[8] Jumratjaroenvanit A, Teng-Amnuay Y. Probability of attack based on system vulnerability life cycle. In: Proc. of the 2008 IEEE Int'l Symp. on Electronic Commerce & Security. 2008. 531−535. [doi: 10.1109/ISECS.2008.212]

[9] Joh H, Malaiya YK. A framework for software security risk evaluation using the vulnerability lifecycle and CVSS metrics. In: Proc. of the 2010 Int'l Workshop on Risk & Trust in Extended Enterprises. 2010. 430−434.

[10] Shahzad M, Shafiq MZ, Liu AX. A large scale exploratory analysis of software vulnerability life cycles. In: Proc. of the 2012 IEEE Int'l Conf. on Software Engineering. 2012. 771−781. [doi: 10.1109/ICSE.2012.6227141]

[11] Cox JLAT. Some limitations of "risk=threat×vulnerability×consequence" for risk analysis of terrorist attacks. Risk Analysis, 2008, 28(6):1749−1761. [doi: 10.1111/j.1539-6924.2008.01142.x]

[12] Mkpong-Ruffin I, Umphress D, Hamilton J, Gilbert J. Quantitative software security risk assessment model. In: Proc. of the 2007 ACM Workshop on Quality of Protection. 2007. 31−33. [doi: 10.1145/1314257.1314267]

[13] Fu ZY, Gao L, Sun Q, Li Y, Gao N. Evaluation of vulnerability serverity based on rough sets and attributes reduction. Chinese Journal of Computer Research & Development, 2016,53(5):1009−1017 (in Chinese with English abstract).

[14] Houmb SH, Franqueira VNL, Engum EA. Quantifying security risk level from CVSS estimates of frequency and impact. Journal of Systems & Software, 2010,83(9):1622−1634. [doi: 10.1016/j.jss.2009.08.023]

[15] Zhao L, Li JE, Lu TB, Liu KP. Research on quantitative assessment model on vulnerability risk for information system. Chinese Journal of Communication, 2009,30(2):71−76 (in Chinese with English abstract).

[16] Li S, Tryfonas T, Russell G, Andriotis P. Risk assessment for mobile systems through a multilayered hierarchical bayesian network. IEEE Trans. on Cybernetics, 2016,46(8):1749−1759. [doi: 10.1109/TCYB.2016.2537649]

[17] Fonseca J, Seixas N, Vieira M, Madeira H. Analysis of field data on web security vulnerabilities. IEEE Trans. on Dependable & Secure Computing, 2014,11(2):89−100. [doi: 10.1109/TDSC.2013.37]

[18] Wiik J, Gonzalez JJ, Lipson HF, Shimeall TJ. Dynamics of vulnerability-modeling the life cycle of software vulnerabilities. In: Proc. of the 22th Int'l Conf. on System Dynamics. 2004. 3043−3061.

[19] Ciapessoni E, Cirio D, Kjølle G, Massucco S, Pitto A, Sforna M. Probabilistic risk-based security assessment of power systems considering incumbent threats & uncertainties. IEEE Trans. on Smart Grid, 2016,7(6):2890−2903. [doi: 10.1109/TSG.2016.2519239]

[20] Li ZJ, Zhang JX, Liao XK, Ma JX. Survey of software vulnerability detection techniques. Chinese Journal of Computers, 2015,38(4):717−732 (in Chinese with English abstract).

[21] Wu SZ. Review and outlook of information security vulnerability analysis. Chinese Journal of Tsinghua University (Science and Technology), 2009,49(s2):2065−2072 (in Chinese with English abstract).

[22] Mell P, Scarfone K, Romanosky S. Common vulnerability scoring system. IEEE Security & Privacy, 2006,4(6):85−89. [doi: 10.1109/MSP.2006.145]

[23] NIST. National vulnerability database. 2018. https://nvd.nist.gov/

[24] CVE. Common vulnerabilities and exposures. 2018. http://cve.mitre.org/

[25] OSVDB. The open source vulnerability database. 2018. http://osvdb.org/

[26] Seneta E. Non-Negative Matrices and Markov Chains. 2nd ed., New York: Springer Science & Business Media, 2006. 128−132.

[27] Beattie S, Arnold S, Cowan C, Wagle P, Wright C. Timing the application of security patches for optimal uptime. In: Proc. of the 16th USENIX Conf. on System Administration. 2002. 233−242.

[28] Symantec. 2017. https://www.symantec.com/connect/ru/blogs/what-you-need-know-about-wannacry-ransomware

附中文参考文献:

[1] 陈恺,冯登国,苏璞睿,聂楚江,张晓菲.一种多周期漏洞发布预测模型.软件学报,2010,21(9):2367−2375. http://www.jos.org.cn/1000-9825/3626.htm [doi: 10.3724/SP.J.1001.2010.03626]

[2] 聂楚江,赵险峰,陈恺,韩正清.一种微观漏洞数量预测模型.计算机研究与发展,2011,48(7):1279−1287.

[3] 高志伟,姚尧,饶飞,刘延钊,罗平.基于漏洞严重程度分类的漏洞预测模型.电子学报,2013,41(9):1784−1787.

[7] 宋明秋,王磊磊,于博.基于生命周期理论的安全漏洞时间风险研究.计算机工程,2011,37(1):131−133.

[13] 付志耀,高岭,孙骞,李洋,高妮.基于粗糙集的漏洞属性约简及严重性评估.计算机研究与发展,2016,53(5):1009−1017.

[15] 周亮,李俊娥,陆天波,刘开培.信息系统漏洞风险定量评估模型研究.通信学报,2009,30(2):71−76.

[20] 李舟军,张俊贤,廖湘科,马金鑫.软件安全漏洞检测技术.计算机学报,2015,38(4):717−732.

[21] 吴世忠.信息安全漏洞分析回顾与展望.清华大学学报(自然科学版),2009,49(s2):2065−2072.

胡浩(1989－),男,安徽池州人,博士生,主要研究领域为网络安全态势感知,网络威胁行为分析,图像秘密共享.

常德显(1977－),男,博士,副教授,主要研究领域为信息安全,可信计算.

叶润国(1976－),男,博士,高级工程师,主要研究领域为大数据安全分析.

刘玉岭(1982－),男,博士,高级工程师,CCF专业会员,主要研究领域为网络安全态势感知,大数据安全分析.

张红旗(1962－),男,博士,教授,博士生导师,主要研究领域为网络安全,风险评估,等级保护,信息安全管理.

杨英杰(1971－),男,博士,教授,主要研究领域为数据挖掘,态势感知,信息安全管理.